Technical Information

TI 34P02K03-03

STARDOM システム解析ガイド
（情報解析編）

YOKOGAWA
横河電機株式会社

TI 34P02K03-03
2012.09.14 11 版（YK）

Blank Page

はじめに

本書は，STARDOM を使用したシステム運用時の障害解析方法のガイドで，STARDOM を使ったシステムをエンジニアリング，保守される方を対象に書かれています。
また，運用後の障害を回避するシステム設計についても説明しますので，システム設計フェーズでも利用することができる内容になっています。
本書は，Windows 2000 Professional/Windows XP Professional/Windows Vista Business Edition/Windows 7 Professinal オペレーティングシステム（以下 Windows 2000/XP/Vista/7 と記述します）のシステム管理についてと VDS/ASTMAC のシステム管理／保守，FCN/FCJ のシステム管理／保守についての知識を持っている方を対象としています。

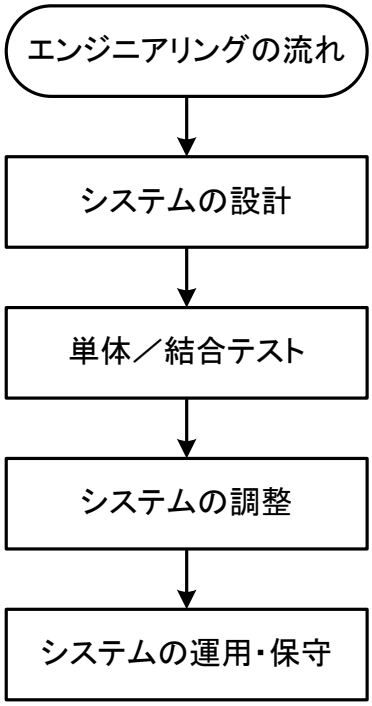


図　STARDOM システム作業の流れ

通常はこのような流れでエンジニアリングを行いますが，本書では「システム設計」での注意する点と「システム調整」時に確認すること，「システム運用・保守」時の障害解析について説明します。
また，VDS/ASTMAC（R5.30 以降）FCN/FCJ（R.1.70 以降）について記述しています。

本書は，システム運用後の障害を避けることを目的としたアプリケーション設計とシステム調整時とシステム運用時の障害対応に利用するものです。

All Rights Reserved. Copyright © 2001, Yokogawa Electric Corporation

■ 本書の構成

本書の構成を以下に示します。

1 章　VDS/ASTMAC 共通

VDS/ASTMAC システム共通したシステム解析について説明します。

2 章　ASTMAC システム

ASTMAC システム固有のシステム解析について説明します。

3 章　FCN/FCJ システム

FCN/FCJ システム固有のシステム解析について説明します。

障害が発生したときの情報取得方法につきましては，「STARDOM システム解析ガイド（情報収集編）TI 34P02K03-02」を参照ください。

著作権および商標

■ 著作権

本書の著作権は弊社に帰属します。
本書をコピーしたり，第三者に譲渡，販売，頒布（パソコン通信のネットワークを通じて通信により提供することを含みます）することを禁止します。また，無断でビデオテープその他に登録，録画することも禁止します。

■ 商標ならびにライセンスソフトウェアについて

- STARDOM は，横河電機株式会社の商標です。
- その他，本文中に使われている会社名・商品名は，各社の商標または登録商標です。

・本文中の各社の登録商標または商標には，TM，®マークは表示しておりません。

Blank Page

STARDOM システム解析ガイド（情報解析編）

TI 34P02K03-03　11 版

目　次

はじめに i
著作権および商標 iii
目　次 v
1. VDS/ASTMAC共通 1
1.1 システム設計の注意点 1
1.1.1 システム運用後に発生する障害 1
1.1.2 メモリ設計 2
1.1.3 ISAM 3
1.1.4 シグナル交信オブジェクト 4
1.1.5 タイマ 5
1.1.6 市販ソフトの利用 6
1.2 システム調整時のチェック 7
1.2.1 測定ツール 7
1.2.2 チェック項目 20
1.3 システム運用時の障害解析 23
1.3.1 システム障害解析方法 23
1.3.2 メモリリークの解析方法 25
2. ASTMACシステム 29
2.1 システム設計の注意点 29
2.1.1 メモリ使用量に関する注意点 29
2.1.2 I/Oアクセス 30
2.1.3 データサーバのVBAとアプリケーションフォーム 34
2.1.4 ブロックデータオブジェクト 36
2.1.5 その他 38
2.2 システム運用時の障害解析 39
2.2.1 システム障害解析方法 39
3. FCN/FCJシステム 41
3.1 レビジョン情報の解析 41
3.1.1 使用するPCのOS（Windows）とバージョン 41
3.1.2 FCN/FCJと開発/保守ツールのレビジョン 42

3.2 LEDの状態の解析 43
3.2.1 電源モジュールのLEDの状態 43
3.2.2 FCN/FCJの稼働状態LEDの解析 45
3.2.3 FCN/FCJの通信状態LEDの解析 49
3.2.4 I/Oモジュールの状態表示LEDの解析 50
3.2.5 SBバスリピートモジュールの状態表示LEDの解析 51
3.2.6 シリアル通信モジュールの状態表示LEDの解析 52
3.2.7 CANopen通信モジュールの状態表示LEDの解析 53
3.2.8 PROFIBUS-DP通信モジュールの状態表示LEDの解析 54
3.3 CPUの状態の解析 55
3.3.1 制御タスクの実行時間 55
3.3.2 CPU負荷率の解析 57
3.3.3 ウォッチドックタイムエラー 59
3.3.4 CPUオーバーロード 61
3.4 ロジックデザイナの情報 63
3.4.1 システム定義のグローバル変数の情報 63
3.4.2 POUのエラー情報 65
3.4.3 ロジックデザイナのログ情報の解析 66
3.5 リソースコンフィギュレータの情報の解析 67
3.5.1 CPUモジュールの基本設定 67
3.5.2 システムカードのライセンス情報 68
3.5.3 RAS情報の解析 70
3.5.4 プロジェクトの情報 72
3.5.5 リソースコンフィギュレータの定義情報 73
3.5.6 ループチェックツールの情報の解析 74
3.6 メンテナンスページの情報の解析 76
3.6.1 Top Pageの情報の解析 76
3.6.2 FCN/FCJのログの解析 76
3.6.3 FCN/FCJの日付と時刻情報の解析 79
3.7 リモートシェルツール（FcxRsh）を用いた解析 80
3.7.1 リモートシェルツール（FcxRsh）の説明 80
3.7.2 リモートシェルツール（FcxRsh）の取得 80
3.7.3 リモートシェルツール（FcxRsh）の実行 81
3.8 FCN/FCJの障害の解析事例 86
3.8.1 リテインデータに関する障害の解析事例 86
3.8.2 入出力信号に関する障害の解析事例 88
3.8.3 時間計測に関する障害の解析事例 89
3.8.4 Heart beat timeoutエラー 90
3.8.5 MISCMPエラー 92
3.8.6 FCN/FCJとPC間の通信ができなくなった。 95

付録A.　パフォーマンスデータ 99
付録A.1 メモリ見積もり 99
付録A.2 通信時間見積もり 100
Technical Information　改訂情報 i

Blank Page

1. VDS/ASTMAC 共通

1.1 システム設計の注意点

1.1.1 システム運用後に発生する障害

Windows 2000/XP/Vista/7 上で動作する VDS/ASTMAC システムを構築した場合に単体テストや結合テストでは見つからずにシステム運用後に発覚する障害があります。障害の要因として Windows 2000/XP/Vista/7 システムの資源を際限なく使ってしまったために発生することが多くあります。したがってシステム設計時に十分に検討すれば防げるものも多くあります。以下にシステム運用後に発生する障害（状況）の例を示します。

● メモリ不足

物理メモリ以上のメモリをシステム全体で使ってしまっており，スワップが頻繁に発生している状態です。システムのレスポンスが遅く，データ収集が正常に行われないなどの障害が発生する可能性があります。

● 仮想メモリ不足

メモリ不足からさらに進んで仮想メモリも不足している状態です。システムが「仮想メモリ不足」のダイアログウィンドウを表示します。システムのレスポンスが非常に遅かったり，システムハングアップやシステムダウンになる場合もあります。

● パフォーマンストラブル

CPU 負荷が高く，画面のレスポンスが遅かったり，データ収集が正常に行われなかったりします。

● データ抜け

ごくまれなタイミングでデータが抜けたり，API がエラーになったりすることがあります。

● 通信エラー

連続運転中に通信でエラーが発生することがあります。

本章では，これらの障害がシステム運用後に発生しないように設計時に注意する点や，（本来あってはならないのですが）問題が発生した場合の原因追究のためにシステムに組み込んでおくべきことを説明します。

1.1.2　メモリ設計

■ Windows 2000/XP/Vista/7 のメモリ管理

Windows 2000/XP/Vista/7 は仮想メモリという機能を使って合計の使用メモリが物理メモリを超えるような場合でもシステムが動作できるようにしています。プログラムから要求されたコードやデータのページがワーキングセット（プログラムから参照できる物理メモリ内のページ）の中にない場合に，ページフォルトが発生し，ディスクページングファイルから読み込みます。ディスクを主記憶の一部のように見せかけることによって，実装されている容量以上のメモリをプログラムが利用可能にする仮想メモリシステムを備えています。ただしコンピュータのメモリが不足してくると，コードおよびデータへのアクセスがディスクに対して頻繁に実行されることになり，システムのボトルネックになる可能性があります。

● 用語

- **WorkingSet（ワーキングセット）**
  プログラムから参照できる物理メモリ内のページ。タスクマネージャのプロセスタブで表示される“メモリ使用量”はワーキングセットです。
- **仮想メモリページングファイル**
  物理メモリからページアウトされた情報が格納されるファイルで［システムボリューム］¥pagefile.sys です。このファイルはなければ Windows 2000/XP/Vista/7 が自動的に作成します。メモリが足りずにこのファイルを使い果たすと最大サイズに達するまで拡張します。サイズはコントロールパネルの「システム」プロパティから変更できます。
- **使用可能メモリ**
  システムの使用可能メモリがある値より小さくなると Windows 2000/XP/Vista/7 は各プロセスのワーキングセットを減らします。結果的にページフォルトが発生し始めます。使用可能バイトの最小値はシステム環境により Windows 2000/XP/Vista/7 が自動的に計算しています。

Windwos 2000/XP/Vista/7 がメモリ管理のために予約するページプールは，物理メモリやアプリケーションからの要求により変化します。よって Windows 2000/XP/Vista/7 に VDS/ASTMAC をインストールした初期状態のメモリ使用量は，それぞれのシステムによって違ってきます。

● 定時起動のプログラムに注意

帳票などのプログラムが起動されているタイミングでメモリ消費量が増えることも考慮に入れてください。その瞬間にスワップが起き，システムレスポンスが低下し，思わぬトラブルが発生したり仮想メモリ不足になることもあります。

● 連続運転を行いメモリリークがないかをチェック

メモリリークは，連続運転を行いメモリ使用量をチェックしないと判明しません。メモリリークは Windows 2000/XP/Vista/7 のバグ，市販ソフトのバグ，アプリケーションのミスなどにより発生する可能性があります。システム調整時に必ずチェックするようにしてください。チェック方法については「1.2.2 チェック項目」を参照してください。

1.1.3 ISAM

● インデックスキーは最小限に

インデックスキーは検索に便利なものですが，多くのキーを付けると書き込み時に処理時間がかかることがあります。必要最低限にするようにしてください。

● 複合インデックスを効率よく使う

あるキーで検索して別の列でソートしてデータを取得したいときは，複合インデックスを使うと効率的にアクセスできます。
例えば，部品コードで検索し，棚番号でソートしてレコードデータを取得したい場合を説明します。
主キーとして部品コード，従属列として棚番号を持つ複合キーを設定します。
ADO の RecordSet オブジェクトの Open メソッドで従属列を条件として省略するとその列に対してソートされて検索されます。

［コーディング例］

```
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset
cn.Provider =“YfsSQL”
cn.Open (“C:¥WORK¥ISAM”)
rs.Open “Select * From tana Where 部品コード=’SD01’”, cn,adOpenForwardOnly, adLockOptimistic, -1
```

ISAM の複合インデックス機能が働き，棚番号のデータ値が昇順になってレコードを取得できます。
部品コードのみのインデックスでは，取得の順番は棚番号について不定になりますので，自分でソートする，または Sort プロパティを使うことになり，対象レコードが多いと処理速度が遅くなります。

● ADO の注意点

- **Sort プロパティよりはインデックスを使う**
  レコードのソートは，ISAM ファイルにインデックスを付与し CommandText でその列を指定して，すぐに実施したほうがはるかに効率的です。ADO の Sort プロパティを使用せずに，インデックスで対応するようにしてください。Sort プロパティは ISAM ではなく ADO 側でソートするので処理時間がかかります。
- **Filter プロパティよりはインデックスを使う**
  Filter は，open で検索したデータから ADO がフィルタするので ISAM のインデックスが働きません。よってディスクアクセスが多数発生する可能性があります。Filter は必ずインデックスを利用した検索結果に対しての二次的な振り分けとして利用ください。
- **RecordCount プロパティに注意**
  RecordCount プロパティは，条件に合ったレコード数を取得するのに使われます。
  例えば RecordRecordset オブジェクトでファイル名のみを指定して Open（全件検索）し，RecordCount プロパティを参照すると総レコード数が取得できます。ただし，この方法は，ADO 内部で全レコードを読み出す方式なので，あまりに多いレコード数（数万件など）では，時間がかかります。この点を考慮して使用してください。

1.1.4　シグナル交信オブジェクト

● シグナル抜けに注意

シグナル交信オブジェクトは，1 オブジェクトあたり 7 個のシグナルをキューイングすることができます。
キューが満杯の状態で送られてきたシグナルは，キューに格納されず捨てられます。
キューが満杯でシグナルがキューにキューイングされなかった場合に，エラーは発生しません。
したがって，シグナル送信と受信処理のバランスを考えてキューがたまることがないような設計を行う必要があります。もしくはブロックデータなどを使用して別途キューを作成してシグナルをトリガとして利用してください。

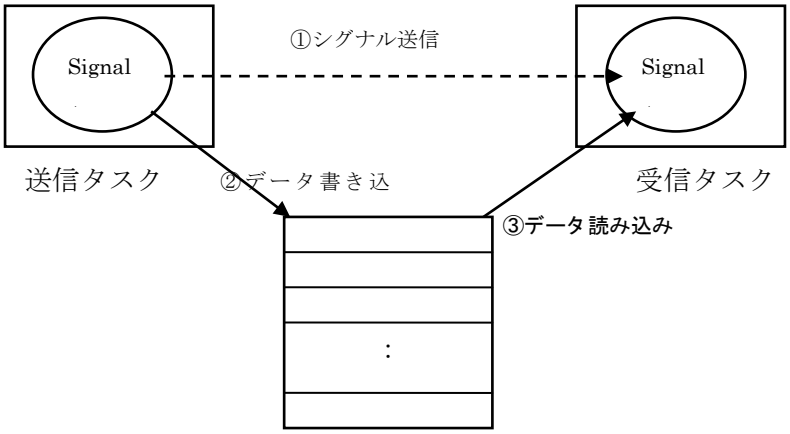


図　シグナル交信オブジェクトの利用

● 複数プロセスからのシグナル送信に注意

1 つのシグナル交信オブジェクトに複数のシグナルがまったく同時に送信されると，送信側で送信ビジーのエラーが発生します。したがって，シグナルの送信には必ずリトライ処理を行ってください。
通常シグナルを受け取るために必要な処理時間は，数 10 ミリ秒ですのでエラーを受け取ってすぐにリトライすることで送信が可能になります。エラーの種類によりリトライが必要なものとそうでないものがあります。リトライが必要なエラーについては「マルチタスク支援パッケージ説明書」（IM 34P02H04-01）「Appendix B エラー一覧」を参照ください。

1.1.5 タイマ

● YfsClock クロックコントロールの利用

VDS/ASTMAC で提供している YfsClock コントロールは継続タイマ以外にワンショットタイマや時刻指定タイマ機能があります。
YfsClock 以外のタイマイベントの中でデータサーバのデータを無手順アクセスするようなコーディングをすると，場合によってオートメーションエラーが発生する場合がありますので，基本的には YfsClock コントロールを使用してください。
なお，サードパーティが提供する Clock 関係のコンポーネントを使用される場合があると思いますが，そちらを使用される場合は，メモリリークなどの不具合がないことを十分確認された上で使用されることをお勧めします。

● タイマ利用上の注意点

1. VB と VDS/ASTMAC（YfsClock）のタイマイベントの違いについて

定周期割り込みのタイマイベントで負荷のかかる処理を行ったとき，VB では設定周期以内に処理が終わらないと次のタイマイベントは捨てられますが，VDS/ASTMAC ではタイマイベントが捨てられることはありません。イベントバッファにたまっていきます。定周期割り込みのタイマイベントで行う処理は設定周期内で十分に完了できる処理としてください。

2. 定周期割り込みタイマの複数使用について

1 つのアプリケーションで複数の定周期割り込みタイマを使用したとき，設定周期が短かいと個々の定周期割り込みのタイマイベントで行う処理が，設定周期内で十分に完了できる処理だったとしても，同時に定周期割り込みのタイマイベントが複数発生すればイベントごとの処理は，設定周期内に完了できるとは限りません。不要なタイマは"Enable=False"にしておき必要なとき"Enable=True"とするべきです。または，複数のアプリケーションに分けて特定のアプリケーションで複数のタイマを同時に使用しないようにします。

3. MsgBox 表示中はタイマが停止する

これは VBA の仕様です。MsgBox 表示中はタイマコントロールは停止し MsgBox 終了時からタイマが再スタートします。タイマを使用する場合は MsgBox を使用しなでください。
同様に VBA のデバッグ中にブレークすると，タイマがリセットされます。

1.1.6　市販ソフトの利用

● ReadMe を読む

必ず利用する前に市販ソフトに付属する ReadMe を読むようにしましょう。
VDS/ASTMAC はもとより，外部のコントロールをどうしても利用しなければならない場合は，しっかり ReadMe を読んでおくのが肝要です。
よく知られる ReadMe タイプのファイルには次のものがあります。

1. **ReadMe.txt**
   制限事項やリリース直前に行った仕様変更，前バージョンからの修正が記載されています。
2. **Regisit.txt**
   レジストリ登録に関係した内容が記載されています。うまく動作していたアプリケーションが突然動かなくなったりしたら，レジストリが破壊を疑う必要があります。そんな場合に，このファイルが生きてきます。
3. **Limit.txt**
   制限事項が多い場合，ReadMe ファイル以外にこのファイルを起こして，詳細情報を記載しているケースがあります。もしこのファイルを見つけたら，要注意です。そういう製品には，複雑な処理をさせないのが賢明です。

● メモリリークの注意

一般的に市販のソフトは連続運転を想定した検証がなされていない可能性が高いと思ってください。よって実操業前にメモリリークをチェックするようにしてください。

1.2 システム調整時のチェック

単体テストそして結合テストが終わった後のシステム調整時に，メモリ使用量やパフォーマンスをチェックしてください。ここでのチェックがシステム運用後に問題を起こさないためにも重要です。
ここではまず測定ツールの説明をします。その後でチェックするポイントをあげて示します。

表　システム解析ツール一覧

ツール	内容
タスクマネージャ	システムの CPU 負荷やメモリ使用量，プロセスの状態を参照するツールです。
パフォーマンスモニタ	Windows 2000/XP/Vista/7 システムの各種パフォーマンスを詳細に確認するツールです。
イベントビューア	Windows 2000/XP/Vista/7 で発生する各種イベントを参照するツールです。
IoInfo	I/O アクセスに関する統計情報を参照するツールです。
IoStat	コントロールオブジェクトの I/O アクセスに関する統計情報を参照するツールです。
IoView	データサーバのキャッシュの現在値を参照するツールです。
opctest	OPC サーバと接続を確認する OPC クライアントツールです。
オブジェクトビュア	データサーバ上の I/O アクセス状況，取得しているデータ値を確認するツールです。

1.2.1 測定ツール

■ タスクマネージャ

システムの CPU 負荷やメモリ使用量，プロセスの状態を参照するツールです。

● 起動方法

開発時などはタスクバーから起動できます。

1. タスクバーの何もない部分をマウスの右ボタンでクリックします。
2. ［タスクマネージャ］をクリックします。

運用時にタスクバーを隠している場合は「CTRL」+「Shift」+「ESC」を押すと「タスクマネージャ」を起動できます。

● 表示内容

- **［アプリケーション］タブ**
  コンピュータで現在実行されているプログラム（タスクとも呼びます）の状態が表示されます。
- **［プロセス］タブ**
  コンピュータで現在実行されているプロセスに関する情報が表示されます。プロセスに関するほかの情報を表示するには，［表示］メニューの［列の選択］をクリックします。
- **［パフォーマンス］タブ**
  コンピュータのパフォーマンスに関する次のような情報が動的に表示されます。

表　Windows XP タスクマネージャ プロセスタブ項目 抜粋

項目	説明	パフォーマンスモニタのカウント
イメージ名	プロセスの名前	［インスタンス］ボックスのプロセス名
PID（プロセス識別子）	プロセスに実行時に割り当てられる数値 ID	Process:ID Process
CPU 使用率	プロセスのスレッドが前回の更新以降プロセッサを使用した時間の割合	Process:%Processor Time
CPU 時間	プロセスが開始後使用したプロセッサの総時間（秒単位）	Process:Elapsed Time
メモリ使用量	プロセスが使用したメインメモリの量（KB）	Process:Working Ser
メモリ使用量デルタ	前回の更新以降のメモリ使用状況の変化（KB 単位）。	なし
最大メモリ使用量	プロセスが使用したメインメモリの量の最大値（KB）	Process：Working Set Peak
ページフォルト	データがメモリ内になかったため，このプロセス用にディスクからデータを取得する必要があった回数。この値は，プロセスが開始した時刻からの累計値になる。	なし。PageFault／Sec は，一定時間のページフォルトの割合。
ページフォルトデルタ	前回の更新以降のページフォルトの回数。	なし
仮想メモリサイズ	プロセスが共有するページングファイルのサイズ	Process:Page File Bytes
ページプール	プロセスが使用したページドプール（ユーザメモリ）の容量（KB 単位） ページドプールは，ディスクにページングできる仮想メモリである。すべてのユーザメモリとシステムメモリの一部が含まれる。	Process:Pool Paged Bytes
非ページプール	プロセスが使用したノンページドプール（システムメモリ）の容量（KB 単位） ノンページドプールは，ディスクにページングされないオペレーティングシステムのメモリである。	Process:Pool Non Paged Bytes
基本優先度	プロセスの基本優先度	Process:Priority Paged Bytes
ハンドル数	プロセスのオブジェクトテーブル内のオブジェクトハンドル数	Process:Handle Count
スレッド数	プロセスで動作中のスレッド数	Process:Thread Count

表　Windows XP タスクマネージャ パフォーマンスタブ項目

項目	説明	パフォーマンスモニタのカウント
CPU 使用率	プロセッサが使用されている時間のパーセント	Prcessor:%Processor Time
メモリ使用量	使用されている仮想メモリの容量	Memory：Committed Byte
合計：ハンドル	すべての使用されているオブジェクトハンドル数。	Process：Handle Count：_Total
合計：スレッド	実行スレッド数	Process：Handle Thread：_Total
合計：プロセス	プロセスの数	Object：Processes
物理メモリ：合計	実装されているメモリ	なし
物理メモリ：利用可能	プロセスが使用できる物理メモリの容量。	Memory： Available Bytes
物理メモリ：ファイル	ファイルシステムキャッシュサイズ	Memory：Cache Bytes
コミットチャージ：合計	使用されている仮想メモリの容量	Memory：Committed Byte
コミットチャージ：制限値	ページファイルを拡大せずに全プロセスに割り当てることができる仮想メモリのサイズ	Memory：Commit Limit
コミットチャージ：最大値	使用された仮想メモリの最大値。仮想メモリが拡張されると制限値を超えることができる。	なし
カーネルメモリ：合計	カーネルが使用しているメモリ	Memory：Pool Paged Bytes と Memory：Pool Nonpaged Bytes の合計
カーネルメモリ：ページ	カーネルに割り当てられたページドプールのサイズ。ページドプールはページアウトされる可能性がある。	Memory：Pool Paged Bytes
カーネルメモリ：非ページ	カーネルに割り当てられたノンページドプールのサイズ。ノンページドプールはページアウトされない。	Memory：Pool Nonpaged Bytes

表　Windows Vista / Windows 7 タスクマネージャ プロセスタブ項目 抜粋

項目	説明
CPU 使用率	プロセッサが使用されている時間のパーセント
メモリ	使用されている仮想メモリの容量
イメージ名	プロセスの名前
PID（プロセス ID）	実行中のプロセスを一意に識別する番号
ユーザー名	そのプロセスが実行されているユーザー アカウント
セッション ID	プロセスの所有者を識別する番号。複数のユーザーがログオンしている場合は，各ユーザーに一意のセッション ID が割り当てられる。
CPU 使用率	前回の更新以降にプロセスが CPU を使用した時間の比率。（列見出しには [CPU] と表示される）
CPU 時間	プロセスが開始されて以来使用したプロセッサ時間の合計（秒単位）。
メモリ：ワーキング セット	プライベート ワーキング セット内のメモリ容量と，プロセスが使用しているメモリ容量のうち他のプロセスと共有可能なメモリ容量の合計。
メモリ：ピーク ワーキング セット	プロセスが使用したワーキング セット メモリの最大容量。
メモリ：ワーキング セット デルタ	プロセスが使用したワーキング セット メモリの変化量。
メモリ：プライベート ワーキング セット	プロセスが使用しているメモリ容量のうち，他のプロセスと共有できないメモリ容量だけを表すワーキング セットのサブセット
メモリ：コミット サイズ	プロセスで使用するために予約されている仮想メモリの容量
メモリ：ページ プール	カーネルまたはドライバーがプロセスのために割り当てた，ページング可能なカーネル メモリの容量。ページング可能なメモリとは，別の記憶メディア（ハード ディスクなど）に書き込めるメモリ。
メモリ：非ページ プール	カーネルまたはドライバーがプロセスのために割り当てた，ページング不可能なカーネル メモリの容量。ページング不可能なメモリとは，別の記憶メディアに書き込めないメモリ。
ページ フォルト	プロセスが開始されて以来発生したページ フォルトの回数。
ページ フォルト デルタ	前回の更新以降のページ フォールト数の変化分
基本優先度	プロセスのスレッドがスケジュールされる順序を決定する優先順位
ハンドル	プロセスのオブジェクト テーブルに含まれるオブジェクト ハンドルの数。
スレッド	プロセスで実行中のスレッドの数。
USER オブジェクト	プロセスによって現在使用されている USER オブジェクトの数。USER オブジェクトは Window Manager のオブジェクトであり，ウィンドウ，メニュー，カーソル，アイコン，フック，アクセラレータ，モニター，キーボード レイアウト，および他の内部オブジェクトがこれに該当する。
GDI オブジェクト	グラフィックス出力デバイス用アプリケーション プログラミング インターフェイス（API）のグラフィックス デバイス インターフェイス（GDI）ライブラリのオブジェクトの数
イメージ パス名	ハード ディスク上のプロセスの場所
コマンド ライン	プロセスを作成するために指定されたコマンド ライン全体
データ実行防止	このプロセスに対してデータ実行防止が有効になっているか，無効になっているかを示す。

表　Windows Vista / Windows 7 タスクマネージャ パフォーマンスタブ項目

項目	説明
物理メモリ（MB）：合計	コンピューターに搭載されている RAM の容量
物理メモリ（MB）：キャッシュ済み	最近システム リソースに使用された物理メモリの容量
物理メモリ（MB）：利用可能	プロセス，ドライバー，またはオペレーティング システムですぐに使用可能なメモリの容量
物理メモリ（MB）：空きメモリ	現在使用されていない，有用な情報が含まれないメモリの容量（キャッシュ済みメモリには有用な情報が含まれている）
カーネル メモリ（MB）：ページ	Windows のカーネルと呼ばれるコア部分で使用されている仮想メモリの容量
カーネル メモリ（MB）：非ページ	カーネルで使用されている RAM メモリの容量
システム：ハンドル	プロセスで使用されている一意のオブジェクト識別子の数
システム：スレッド	比較的大きなプロセスまたはプログラム内で実行されているオブジェクトまたはプロセスの数
システム：プロセス	コンピューター上で実行されている個別のプロセスの数 （この情報は [プロセス] タブでも表示可能）
システム：起動時間	コンピューターが再起動されてから経過した時間
システム：コミット（MB） システム：ページ ファイル	仮想メモリ（ページング ファイル）の使用状況の説明。最初の数値は現在使用中の RAM と仮想メモリの容量，次の数値はコンピューターで使用可能な RAM と仮想メモリの容量。
[リソース モニター]	メモリおよび CPU リソースの使用量に関する詳細情報。ディスクの使用やネットワークの使用など，リソースに関する詳細も含まれる。

■ パフォーマンスモニタ

Windows 2000/XP/Vista/7 システムの各種パフォーマンスを詳細に確認するツールです。

● 起動方法

Windows 2000/XP の場合，パフォーマンスモニタは，［スタート］メニューの（［設定］）の［コントロールパネル］を開き，［管理ツール（共通）］の中から起動します。
Windows Vista の場合，「コントロールパネル」（「クラシック表示」）から「管理ツール」内の「信頼性とパフォーマンスモニタ」を起動します。
Windows Vista/7 の場合，「コントロールパネル」（表示方法：「小さいアイコン」）から「管理ツール」内の「パフォーマンスモニター」を起動します。

● 機能説明

パフォーマンスモニタは，ローカルコンピュータ，またはネットワーク上のほかのコンピュータのパフォーマンスを測定するためのグラフィカルツールです。パフォーマンスモニタでは，各コンピュータについて，プロセッサ，メモリ，キャッシュ，スレッド，プロセスなどのオブジェクトの動作を表示できます。それぞれのオブジェクトには，カウンタのセットが関連付けられています。カウンタは，装置の使用状況，キュー（待ち行列）の長さ，遅延，および処理速度と内部負荷の測定に使う情報を示します。
パフォーマンスモニタでは，現在の活動と収集中のログの内容を反映したグラフ，警告，およびレポートを作成できます。また，既存のログファイルを開き，現在の活動と同じように，表示したりグラフ化したりできます。

パフォーマンスモニタでは，次のような操作を行い，オブジェクトのパフォーマンスを表示できます。

1. 任意の数のコンピュータのデータを同時に表示します。
2. 現在の活動を反映し，動的に変化するグラフを表示します。グラフと，グラフに表示されるカウンタ値は，ユーザが定義した間隔で更新されます。
3. グラフ，ログ，警告ログ，およびレポートのデータをスプレッドシートまたはデータベースに書き出し，後から加工または印刷します。
4. オブジェクトに関するデータが入ったログファイルを作成します。
5. 既存の複数のログファイルの一部を単一のファイルに追加することにより，長期間にわたるアーカイブを作成します。
6. 現在の活動のレポートを表示します。または，既存のログファイルからレポートを作成します。
7. グラフ，警告，ログ，レポートの設定を個別に，あるいは，ワークスペース（作業領域）全体の設定を保存し，必要に応じて再使用します。

パフォーマンスモニタの詳細は，オンラインヘルプか Windows 2000 の場合は，『Windows 2000 リソースキット』の『リソースガイド』を Windows XP の場合は，『Windows XP Professional リソースキット上下』を参照してください。

■ イベントビューア

Windows 2000/XP/Vista/7 で発生する各種イベントは逐次イベントログに記録されています。このイベントをイベントビューアで参照することができます。

● 起動方法

イベントビューアは，［スタート］メニューの［設定］の［コントロールパネル］を開き，［管理ツール（共通）］の中から起動します。
Windows Vista/7 の場合は，「コントロールパネル」（「クラシック表示」あるいは表示方法：「小さいアイコン」）から「管理ツール」内の「イベントビューアー」を起動します。さらに左のツリーから「Windows ログ」を選択します。

● 機能説明

イベントビューアはシステムログ，セキュリティログ，アプリケーションログを切り替えて表示するものです。VDS/ASTMAC では問題発生時に，システムログとアプリケーションログの内容を必ず確認してください。

- **システムログ**
  Windows 2000/XP/Vista/7 システムのコンポーネントで発生したイベントがログされます。VDS/ASTMAC ではライセンスキーをアクセスする ID モジュールドライバのイベントがシステムログに記録されます。

- **アプリケーションログ**
  アプリケーションで発生したイベントが記録されます。VDS/ASTMAC からのメッセージはアプリケーションログに記録されます。その内容については「セットアップマニュアル IM 34P02A01-02」を参照ください。

● ログの保存

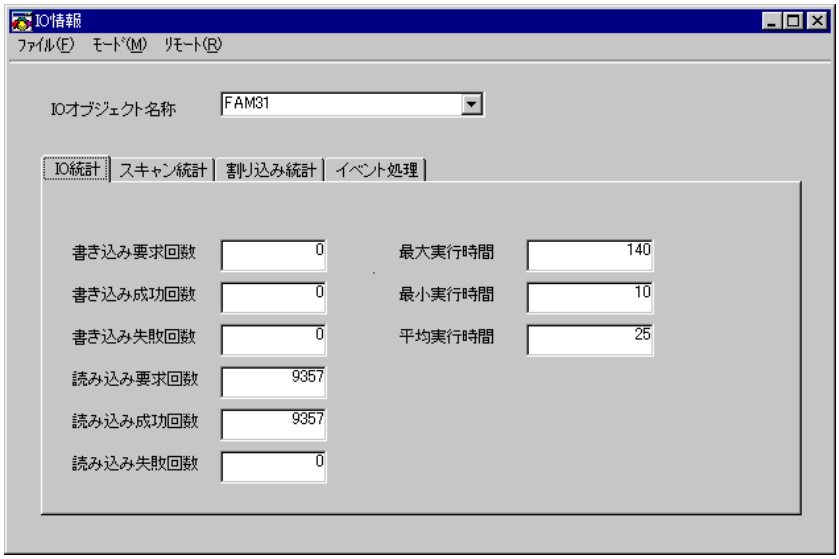
Windows 2000/XP の場合は，保存するログをクリックし，「操作」メニューの「ログファイルの名前を付けて保存」をクリックします。「“選択したログ”の名前を付けて保存」ダイアログが表示されます。ファイルの種類はイベントログファイル形式で保存するかテキスト形式，CSV 形式で保存するかを選択できます。イベントログファイル形式で保存した場合はイベントビューアで開いて参照することができます。（CSV 形式，イベントログファイル形式の両形式で保存しておくことを推奨します。）
Windows Vista/7 の場合は，右クリックメニュー「イベントに名前をつけて保存」あるいは「すべてのイベントを名前をつけて保存」を選択して保存します。

■ IoInfo

I/O オブジェクトの I/O アクセスに関する統計情報を参照するツールです。ただし本ツールは弊社解析用に作られたツールですので，ここでは一般のユーザの方に関連する情報を説明します。

● 起動方法

［VDS/ASTMAC インストールパス］¥System¥DataServer¥Program フォルダにある“IoInfo.EXE”を起動します。
参照したい I/O オブジェクトを I/O オブジェクト名称の横のコンボボックスで選択します。メニューから［モード(M)］－［オンライン］を選択すると現在の状態が表示されます。



● 説明

【IO 統計タブ】

- 書き込み要求回数
  I/O ドライバに対する書き込み要求回数です。
- 書き込み成功回数
  機器への書き込みが成功した回数です。
- 書き込み失敗回数
  機器への書き込みが失敗した回数です。
- 読み込み要求回数
  I/O ドライバに対する読み込み要求回数です。
- 読み込み成功回数
  機器からの読み込みが成功した回数です。
- 読み込み失敗回数
  機器からの読み込みが失敗した回数です。
- 最大実行時間
  機器へのアクセスの最大実行時間です。
- 最小実行時間
  機器へのアクセスの最小実行時間です。

・ 平均実行時間
機器へのアクセスの平均実行時間です。

【スキャン統計タブ】

・ スキャン通算回数
スキャンの通算回数です。
・ オーバーラン発生回数
スキャンが周期内に完了していない回数です。スキャン周期やオブジェクト数，I/O 割り付けなどを調整してください。
・ 最大スキャン時間（ms）
スキャンの最大時間です。複数の周期があるとその中で最大時間が表示されます。各周期ごとの統計情報はありません。
・ 最小スキャン時間（ms）
スキャンの最小時間です。複数の周期があるとその中で最小時間が表示されます。
・ 平均スキャン時間（ms）
平均は，前回値までの平均と，最新値の平均なので，複数周期のスキャンを行っている場合，あまり意味がありません。

補足

スキャン統計は，単純にスキャンに要した時間を計測したものです。複数の周期が重なった場合（例えば，1S/2S/4S が 4S で重なった場合）スキャンの最適化が行われてすべてが同一スキャンで実行されその処理時間が測定されます。したがって各周期ごとの処理時間を知ることはできません。ある周期以外のスキャンを OFF にすれば，最大スキャン，最小スキャン時間は意味のある値となり，その周期のスキャン処理時間を把握することができます。

■ IoStat

コントロールオブジェクトの I/O アクセスに関する統計情報を参照するツールです。ただし本ツールは弊社解析用に作られたツールですので，ここでは一般のユーザの方に関連する情報を説明します。

● 起動方法

［VDS/ASTMAC インストールパス］¥System¥DataServer¥Program フォルダにある“IoStat.EXE”を起動します。
メニューから［モード(M)］－［オンライン］を選択すると現在の状態が表示されます。

● 説明

- **エラーの可能性があるオブジェクト一覧**
  最後のアクセスが，エラーとなっている可能性があるオブジェクトの一覧です。選択したオブジェクトの IO 情報が右側に表示されます。

- **グループ選択とオブジェクト選択**
  任意のオブジェクトを選択できます。グループを選択後にオブジェクトを選択すると IO 情報が表示されます。

- **IO 情報**
  選択されたオブジェクトの IO 情報です。

表　IO 情報

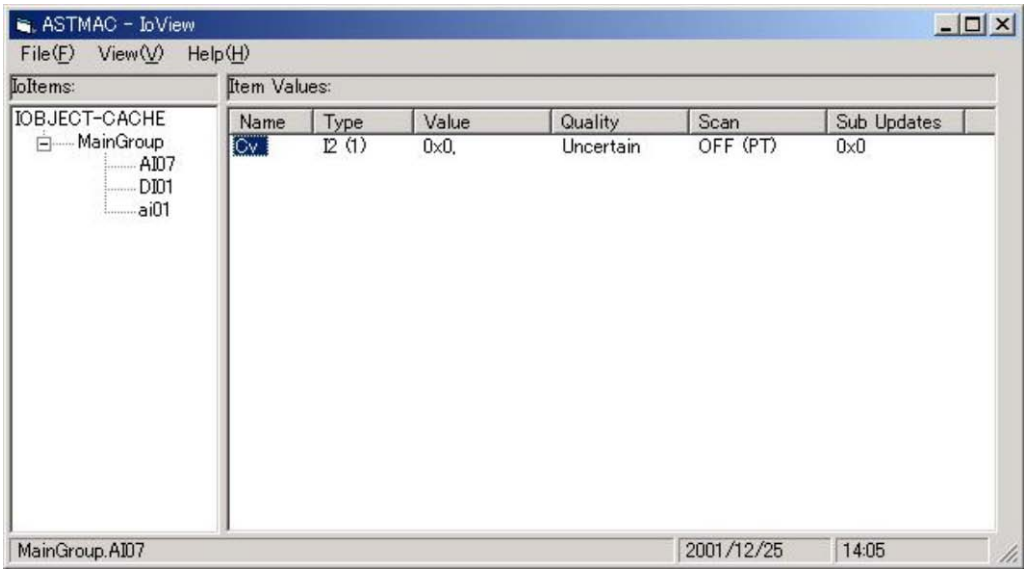
項目	説明
IO 接続状態	接続中：アクセス可能 切断中：I/O アドレスの間違いの可能性があります。
現在のデータ品質	0：正常 1：アラーム 2：無効
最新のエラーコード	9583：I/O オブジェクトの要求キューがいっぱいです。非同期 I/O の頻度を見直してください。 9501：タイムアウト 同時に I/O ドライバがイベントログにエラーを記録しておりますので，そちらも参照ください。 4005：IO エラー（通信/プロトコルなどのエラー） 同時に I/O ドライバがイベントログにエラーを記録しておりますので，そちらも参照ください。
その他	イベントビューアのアプリケーションログを参照してください。

■ IoView

データサーバのキャッシュの現在値を参照するツールです。ただし本ツールは弊社解析用に作られたツールですので，ここでは一般のユーザの方に関連する情報を説明します。なお，このツールは操業モードのときのみ使用してください。

● 起動方法

［VDS/ASTMAC インストールパス］¥System¥DataServer¥IOObject フォルダにある“IoView.EXE”を起動します。
左側のウィンドウの［IOBJECT-CANCHE］をダブルクリックし，各コントロールオブジェクトをクリックするとそのオブジェクトの状態が表示されます。



● 説明

Value
データサーバ内にある生データが表示されます。
Scan
データ更新の状態とスキャン方式が表示されます。
SubUpdates
データ更新を行うたびにカウントアップされます。

注： データサーバのモード切り替えを行う前に，必ず終了させてください。

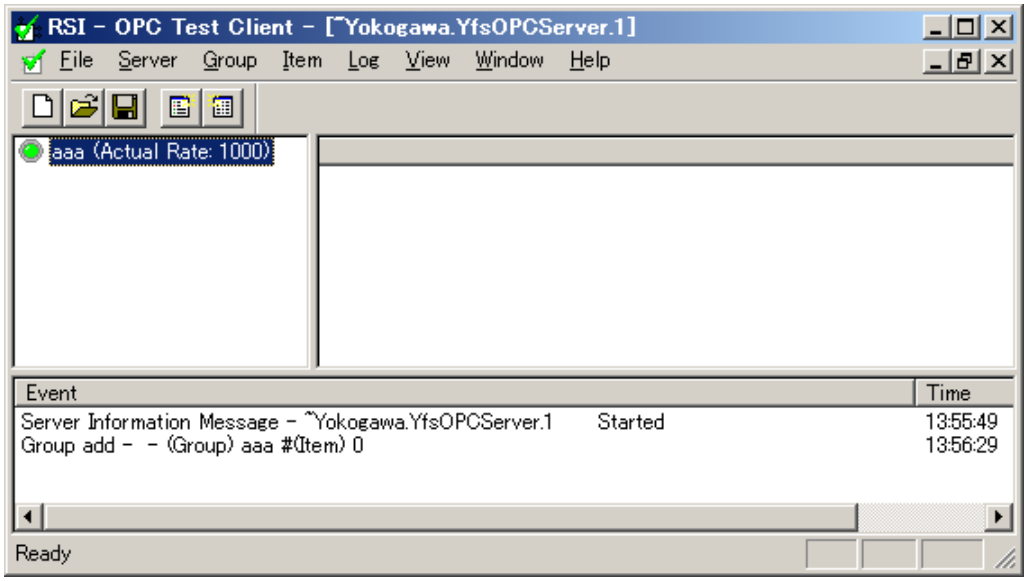
■ OpcTest

OPC サーバの接続状態を確認する OPC クライアントツールです。OPC サーバとの接続がうまく行かないときに，このツールを用いて切り分けを行います。

● 起動方法

［VDS/ASTMAC インストールパス］¥System¥DataServer¥Program フォルダにある“opctest.exe”を起動します。

1. OPC サーバの ProgID を登録
   メニューの［Server］-［Connect］をクリックし，接続したい OPC サーバの ProgID を選択します。（リモートの OPC サーバと接続したい場合は，Node Name に IP アドレスを設定します。）
2. グループ名の登録
   左側のフレーム上にマウスを持っていき，右クリックで Add Group を選択します。適当な Group Name を登録します。（グループ名はなんでもかまいません。）
3. Item 名の登録
   右側のフレーム上にマウスを持っていき，右クリックで Add Item を選択します。Add New OPC Item 設定画面の下にブラウズ機能があるので，そこから確認したい Item を選択します。


RSI - OPC Test Client - [Yokogawa.YfsOPCServer.1]
File Server Group Item Log View Window Help
aaa (Actual Rate: 1000)
Event
Time
Server Information Message - Yokogawa.YfsOPCServer.1 Started 13:55:49
Group add - - (Group) aaa #(Item) 0 13:56:29
Ready

● 説明

- ItemID
  OPC アイテムのユニークな識別子が表示されます。
- SubValue
  サーバから読み出した最新のデータ値が表示されます。
- Sub Quality
  サーバから読み出した最新の品質フラグが表示されます。
- SubUpdates
  データ更新を行うたびにカウントアップされます。

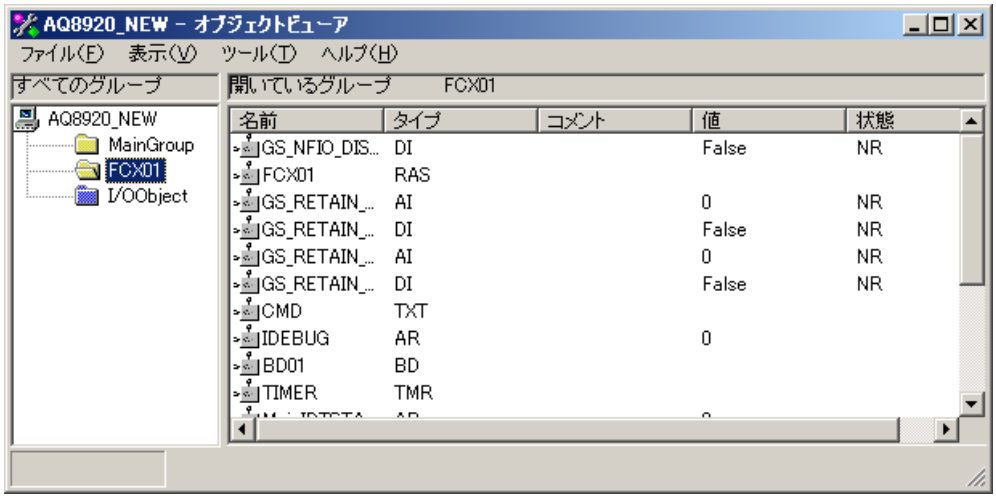
■ オブジェクトビューア

データサーバで接続している I/O のデータ値および，アラームステータス状態を確認するツールです。

● 起動方法

データサーバのモードを「デバッグ」または，「ランタイム」にします。

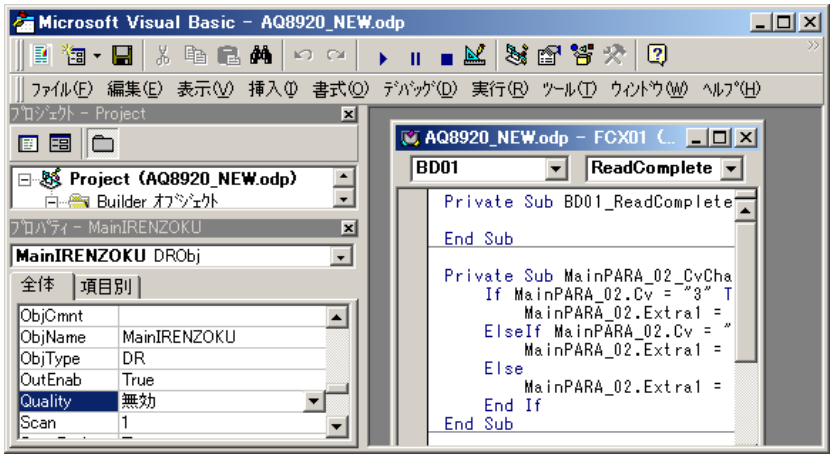
1. データサーバのメニューのツールから「ビューア」を選択します
2. 確認したデータが格納されているグループフォルダを開きます。



● 説明

- 値
  現在格納されているデバイスタグオブジェクトの値が表示されます。
- 状態
  アラームステータスの状態が表示されます。

注　デバイスタグオブジェクトで，「アラーム使用」にチェックが入っている場合のみ，アラームのステータス状態が変化します。
（COMM は通信異常ですが，アラーム使用にチェックが入っていないと，NR で表示されます。通信状態の確認は，データサーバの「VBE」から，プロパティ状態を表示し，Quality で確認してください。）



1.2.2 チェック項目

● パフォーマンスをチェック

まずタスクマネージャの CPU 使用率をチェックしてください。常時 85%を超えるようなシステムはアプリケーションプログラムの見直しか CPU のアップグレードを検討してください。
CPU 使用率の適正値は一概に言えません。場合によっては常時 50%超えるようなシステムも問題である場合があります。CPU 負荷は極力さげるようにお願いします。
次にタスクマネージャの「プロセス」タブで“CPU”を確認してください。この値が高いプロセスが CPU を占有しています。CPU 列のタイトルをクリックすることで CPU 負荷によりソートして表示することができます。

アプリケーションのチェックポイント
- 無駄なループ処理を行っていないか？
- ISAM ファイルの検索に無駄はないか？
- タイマの周期は短すぎないか，タイマ処理の時間が長くないか？

● データ収集が指定時間内に終了しているか？タグ数，スキャン周期に問題はないか？

まずタスクマネージャで，コミットチャージの合計が物理メモリサイズを越えていないことを確認してください。
　コミットチャージ＞＝物理メモリ
となっているシステムはスワップが発生してシステムの応答が遅くなる可能性があります。
さらに詳しくは，パフォーマンスモニタでパフォーマンスオブジェクトの「Memory」で「Pages Reads/sec」カウンタを監視してください。高速な応答を要求するような VDS/ASTMAC システムでは，通常稼働時において 0 であるべきです。

● メモリリークをチェック

連続テストでメモリ不足が発生する場合は，アプリケーションがメモリリークしている可能性があります。
タスクマネージャの「プロセス」タブで仮想メモリサイズを表示して，しばらく連続テストを行ってください。この値が定常的に増加する場合はそのプロセスがメモリリークしています。
「表示」メニューの「更新の頻度」サブメニューの「一時停止」を選択すると更新がとまります。このまま連続テストを行った後で「表示」メニューの「最新の情報に更新」を選択することで最新の情報を表示すると増加しているプロセスを特定しやすくなります。

● 仮想メモリの使用状況をチェック

パフォーマンスモニタでパフォーマンスオブジェクトの「Pageing File」の「% Usage」カウンタと，「%Usage Peak」カウンタを監視してください。これらが100%近くを示している場合，ページングファイルの初期サイズを増やすことを検討してください。ただし，高速な応答を要求するようなシステムでは一時的な対処であり，主記憶を増やすことをお勧めします。

● I/O アクセスのパフォーマンスをチェック

イベントビューアのアプリケーションログに「ソース名：YFSIOOBJECT，イベント ID：2004 スキャンオーバーラン」が記録されてるかをチェックしてください。または IoInfo ツールの「スキャン統計」タブのオーバーラン発生回数を参照してください。
オーバーランが発生している場合は，スキャン周期やオブジェクト数，I/O 割り付けなどを調整してください。

I/O アクセス時間の概要を知るには IoInfo ツールでスキャン実行時間をチェックしてください。
また，これはシステム調整時にやることではありませんが，例えば AI オブジェクトを 1 つ定周期で定義して最大実行時間を見れば PLC に対する 1 要求あたりのパフォーマンスを知ることができます。

● タイマ処理時間をチェック

・ 周期の確認方法

設定周期どおりにタイマイベントの割り込みが発生しているか確認するには Timer プロシージャの中で time 関数をコールして確認します。このとき注意することは，time 関数の表示は 1 秒単位であるのに対してタイマの Interval は 100msec 単位で設定できることです。もし，Interval を 100msec で設定した場合は同時刻に 10 回のタイマイベントが発生します。
下記は，デバッグプリントで VBA のイミディエイトウィンドウにタイマイベントが発生した時刻を表示します。

［記述例］

```
Public pTime As String
Public pCount As Integer
Private Sub YfsClock1_Timer()

   If pTime <> CStr(Time) Then
         Debug.Print "時刻"; pTime; "でのイベント発生回数="; pCount

         pTime = Str(Time)
         pCount = 1
   Else
         pCount = pCount + 1
   End If
End Sub
```

・処理時間の確認方法

プロシージャ内での処理にどのくらいの時間がかかっているかを確認するには，timer 関数を使用します。
下記は，ラベルに現在時刻を 30000 回行ったときの処理時間をデバッグプリントでイミディエイトウィンドウに表示します。TotalTime=1.371094 のときは，この処理に約 1.37 秒かかったことになります。

注： 短い Interval を設定した場合に下記記述例を試したとき，タイマイベントがたまることになります。プログラムの実行停止ができなくなる可能性があります。このような場合は，タスクマネージャでプログラムを終了してください。

［記述例］

```
Private Sub YfsClock1_Timer()
   Dim i As Integer
   StartTime = Timer            'スタートタイム
   For i = 1 To 30000
        Label1.Caption = Time      'ラベルに現在時刻を表示
   Next i
   EndTime = Timer              '終了タイム
   TotalTime = EndTime - StartTime 'トータル
   Debug.Print TotalTime
End Sub
```

注： Timer 関数は，0 時 0 分 0 秒から通算時間です。このため日をまたがるときにこの測定は行えません。

●通信テスト

・接続をチェック

「ping」コマンドを実行して通信テストを行うことができます。「ping」に対する応答が正常に返ってくれば，TCP/IP の通信が正常に機能し，相手ホスト，通信メディア（ケーブル，トランシーバなど）が正しく機能していることが確認できます。なお，自ノードの通信機能のテスト（実際のメディアへの信号送出を除く）を行うことができます。コマンドプロンプトから「ping」を実行してください。

［例］

```
C:¥ VDS/ASTMAC ＞ ping  OPS2
Replay from 192.168.1.202 :bytes=32 time<10ms TTL =128 （正常時）
Bad IP address ViewClient1 （ホスト名登録がなされていない場合）
Request timed out （通信異常）
```

・TCP/IP ネットワークコンフィギュレーションをチェック

ipconfig コマンドでネットワークアダプタのコンフィギュレーション情報を参照できます。

［例］

```
C:¥＞ ipconfig
Windows2000 IP Configuration
Ethernet  adapter  XXXXX
IP Address･････････  192.168.1.1
Subnet Mask･･･････ 255.255.255.0
Default Gateway･･･ 0.0.0.0
```

1.3 システム運用時の障害解析

1.3.1 システム障害解析方法

● 基本的な情報収集

1. タスクマネージャでプロセスの状態，メモリ使用量，CPU 負荷をチェックしてください。
2. イベントビューアでシステムログとアプリケーションログをチェックしてください。
3. メッセージビューア（アプリケーションでログを行っている場合）をチェックしてください。

● システムの応答が遅くなった場合

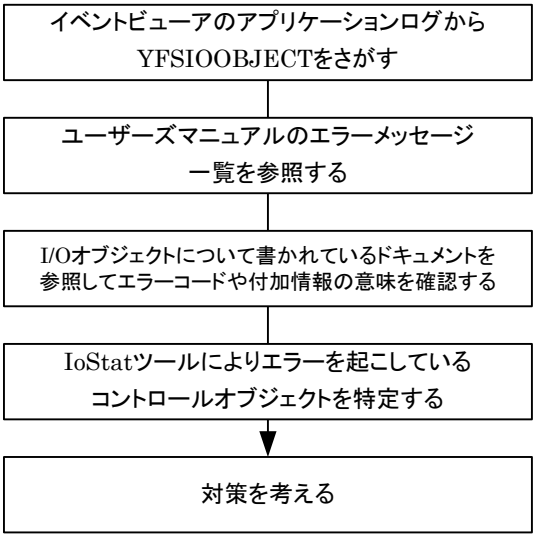
1. タスクマネージャで CPU 負荷をチェックしてください。CPU 負荷が高い場合は「プロセス」タブにおいて“CPU”の値が高いプロセスを特定してください。
2. タスクマネージャで「パフォーマンス」タブのメモリ使用量をチェックしてください。メモリ使用量が以前より極端に増加している場合は，「プロセス」タブの“仮想メモリサイズ”をチェックしてください。極端に増えているプロセスがあるならメモリリークを起こしている可能性があります。プロセスが特定できたなら，どの機能を使うとメモリリークするかを断定しなければなりません。

● マウス操作が効かなくなった場合

Ctrl＋Shift＋ESC キーを押してタスクマネージャを表示させて情報を収集してください。
タスクマネージャが表示されないとき，もし別の Windows 2000/XP/Vista/7 マシンが LAN 上にあるならパフォーマンスモニタとイベントビューアを使い基本的な情報収集を行ってください。

● I/O 装置との通信でエラーが発生した場合

イベントビューアでアプリケーションログをチェックして，ソース名が YFSIOOBJECT である I/O オブジェクトのエラーを探してください。また IoStat ツールによりエラーを起こしているコントロールオブジェクトを特定してください。



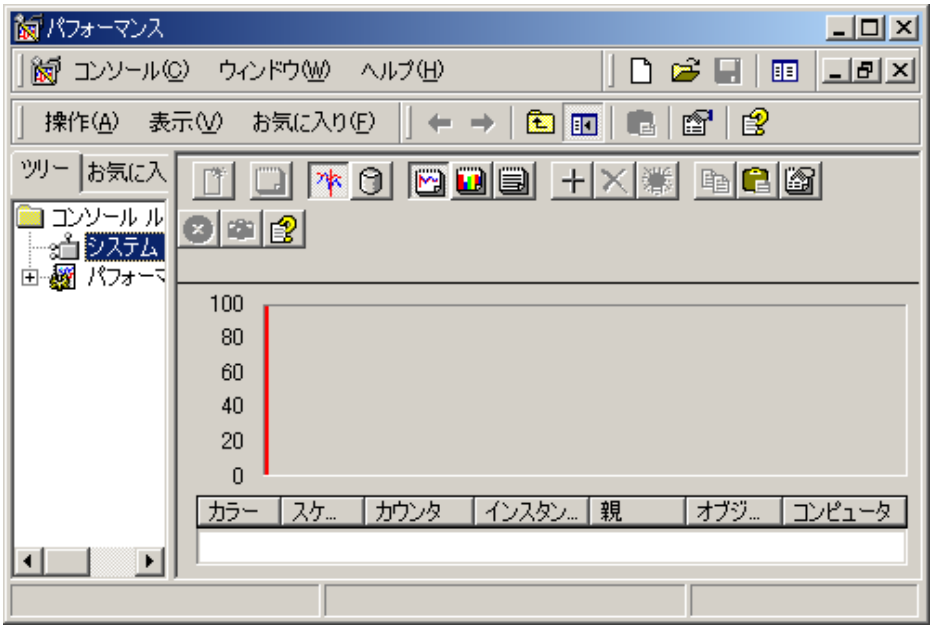
● ブルースクリーンが発生した場合

ブルースクリーンは，BIOS や各種ドライバのバグ，OS のバグ，ハード不良など原因です。まずは，パソコンメーカーにお問い合わせください。
グラフィックドライバやプリンタドライバを最新にすることも検討してください。通常ベンダーのホームページから入手できます。

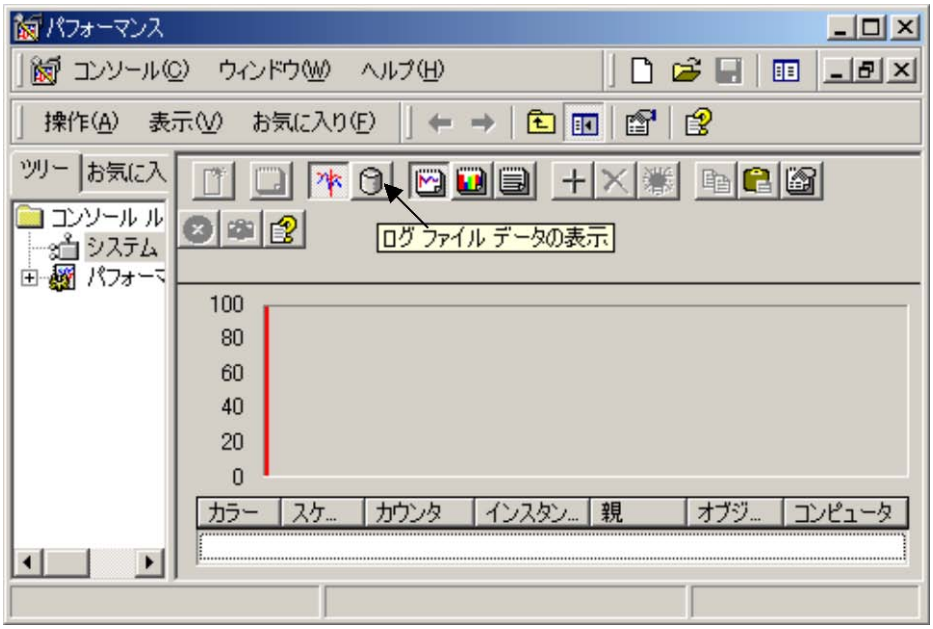
1.3.2 メモリリークの解析方法

● ログファイルの読み込み

1. Windows 2000/XP の場合「コントロールパネル」→「管理ツール」内の「パフォーマンス」を起動します。
   Windows Vista の場合，「コントロールパネル」（「クラシック表示」）から「管理ツール」内の「信頼性とパフォーマンスモニタ」を起動します。
   Windows 7 の場合，「コントロールパネル」（表示方法：「小さいアイコン」）から「管理ツール」内の「パフォーマンスモニター」を起動します。
   以下の画面のイメージは，Windows XP の場合のものです。



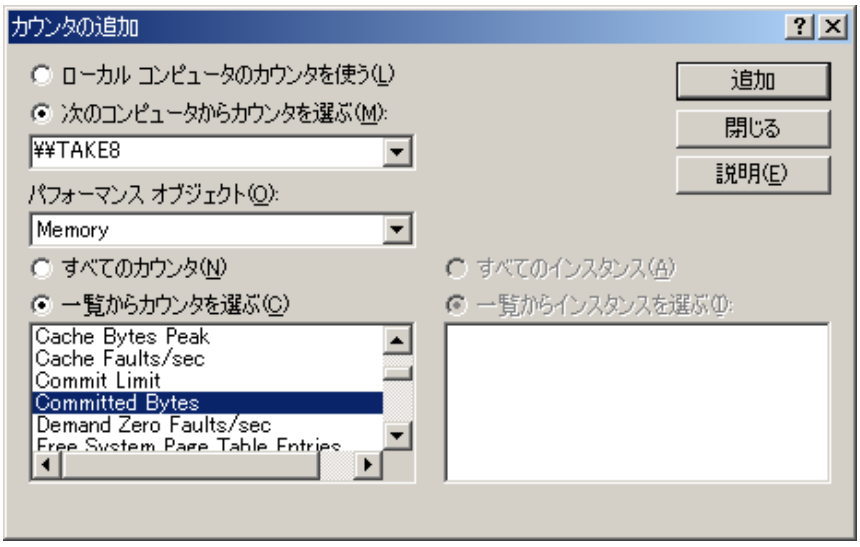
2. 右エリアの「ログファイルデータの表示」あるいは「ログデータの表示」から参照するログファイルを指定して表示します。



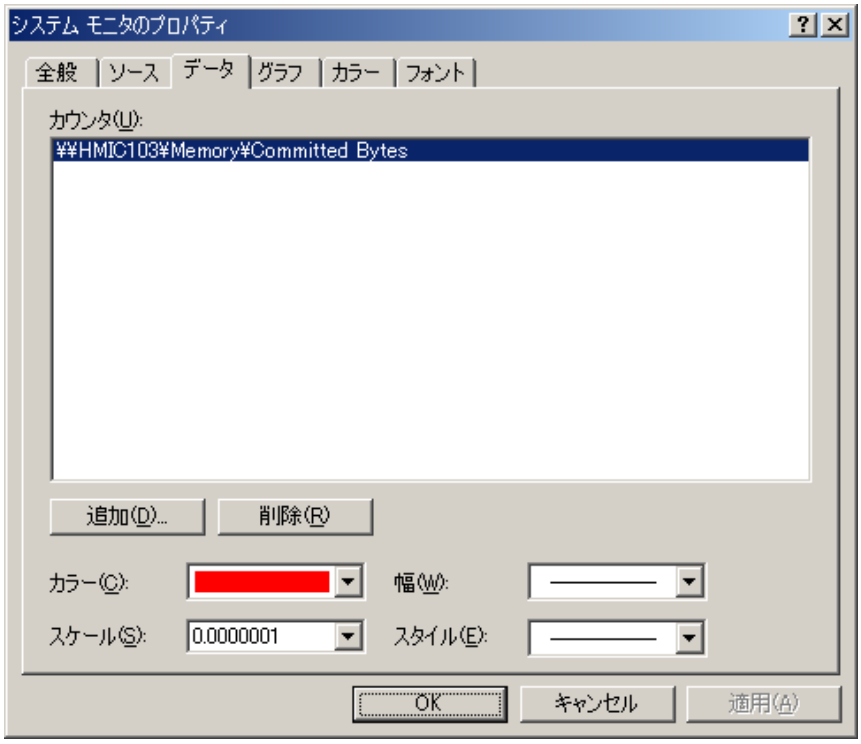
● メモリリーク解析

1. Memory の Committed Bytes の状態を確認します。

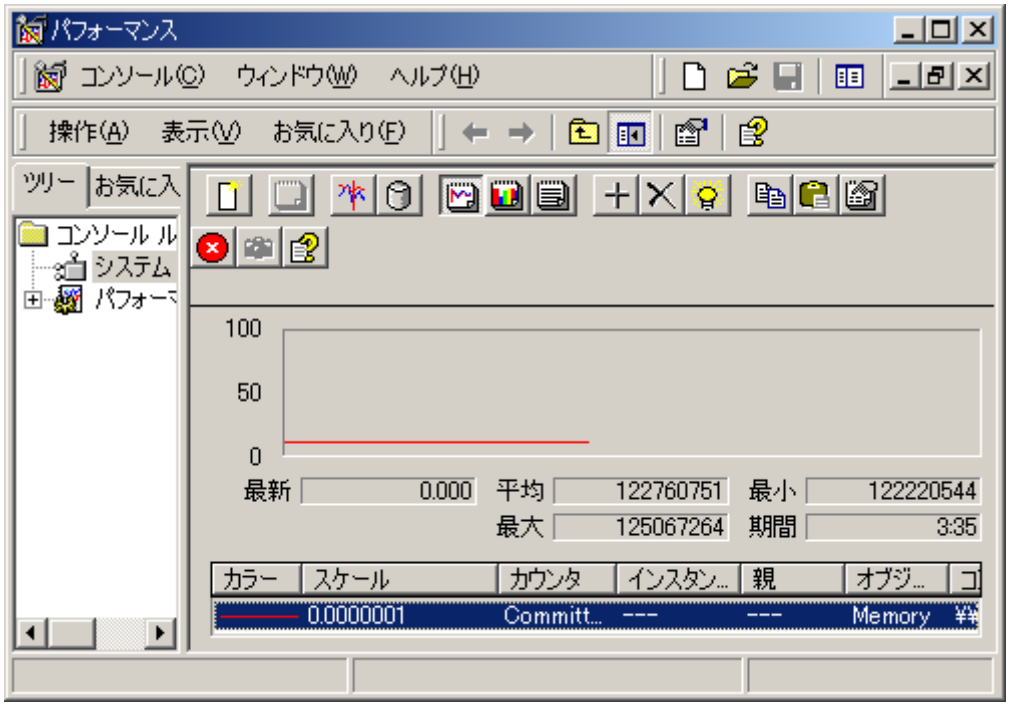
1) 「パフォーマンス（モニター）」の右エリアのグラフ表示部分で，右クリックメニューの「カウンタの追加」を選びます。
2) 「パフォーマンスオブジェクト」あるいは「使用可能なカウンター」で「Memory」を選択します。
3) 一覧から「Committed Bytes」を選択して追加ボタンを押します。
以下，の画面イメージは，Windows XP の場合のものです。



Committed Byte のカウンタが増えていないか確認します。短期間では，多少増加します。アプリケーションシステムにもよりますが，数日間様子を見ることをお勧めします。



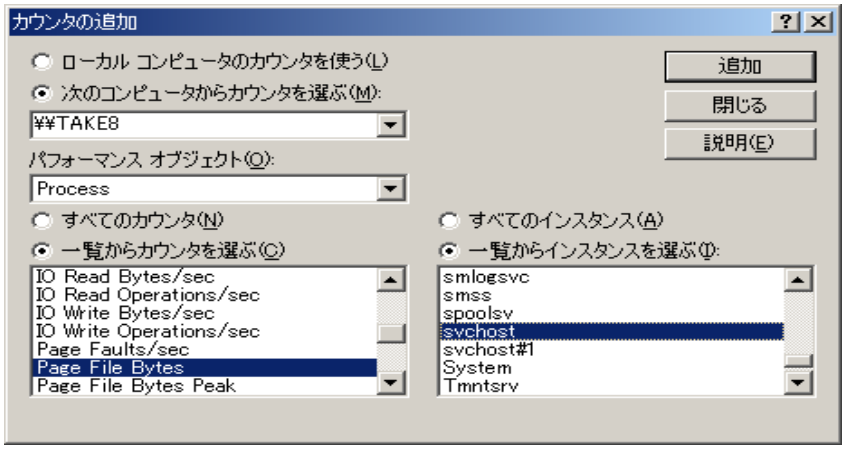
注　カウンタ値がスケール上限で表示されている場合，右クリックメニューで「プロパティ」を選び，「データタブ」でスケールの調整を行ってください。

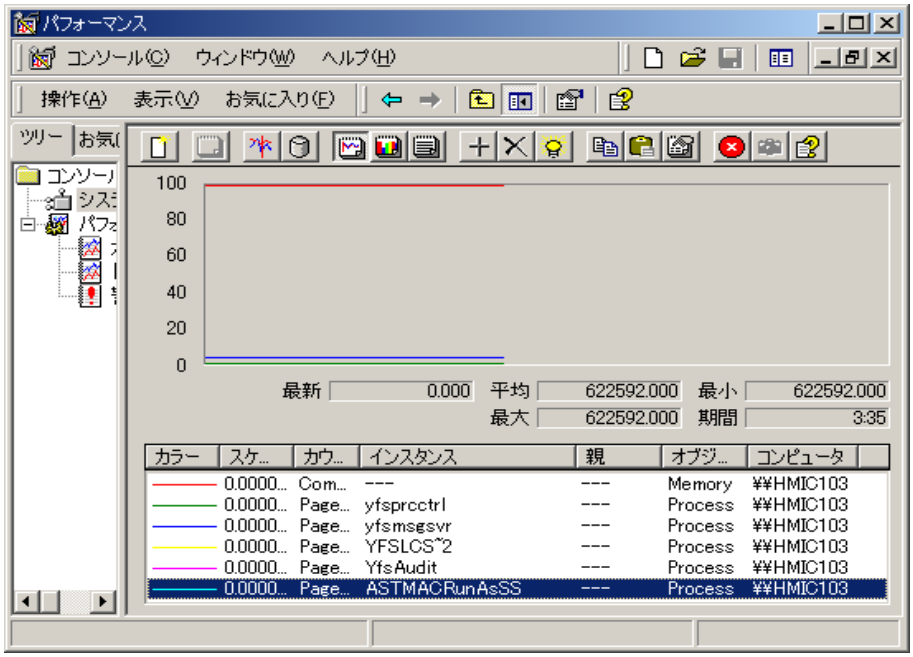


2. Process の「Page File Bytes」の状態を見ながら，リニアに増えているプログラムを探します。短期間では多少増加します。数時間から数日間様子を見ていただくことをお勧めします。

1) 「パフォーマンス」の右エリアのグラフ表示部分で，右クリックメニューの「カウンタの追加」を選びます。
2) 「パフォーマンスオブジェクト」あるいは「使用可能なカウンター」で「Process」を選択します。

3) 一覧から「Page File Bytes」を選択します。
4) 確認するプロセスは以下の順で確認していきます。
4-1)「Yfs～」で始まるプロセスを確認
4-2)そのほかのプロセスを確認
5) 追加ボタンを押します。





2. ASTMAC システム

2.1 システム設計の注意点

2.1.1 メモリ使用量に関する注意点

本節では，メモリ使用量について主にシステム設計時に考慮する点を説明します。
チェック方法については「1.2 章 システム調整時のチェック」を参照ください。
また VDS/ASTMAC が消費するメモリについては「付録 A パフォーマンスデータ」を参照ください。

● アプリケーションフォームの数は最低限に

1 アプリケーションフォームあたり概算ですが下記のメモリを消費します。
4.0＋（0.5[注 1]×ページ数）［MB］に
（データサーバのオブジェクト数+データサーバのオブジェクト数×100）×100［Byte］を加算。

例) 1 画面 label 50 個を貼り付けたページが 10 ページあり，データサーバ上に 200 オブジェクト定義してある場合
4.0+（0.5×10）+（200+200×100）×100×1／1024*1024
となり，約 11M のメモリを消費します。

［注 1］： 1 画面 label 50 個を貼り付けた場合です。実際の表示部品や ActiveX コントロールの種類，数によって変化します。

タスクを分けて処理する必要がない場合は 1 つのアプリケーションフォームが複数のページを持つように設計してください。
ページの終了時（閉じる）に Close するとメモリを解放しますが，反対に次に開くときに表示のコントロールなどをロードするのに時間がかかります。反対に Hide しておくと再表示が格段に早くなりますがメモリは解放されません。メモリ使用とパフォーマンスとのバランスを考慮してください。

● ActiveX コントロールの数は最低限に

1 枚のフォームやページに貼り付けるコントロールの数はできるだけ少なくするほうが賢明です。コントロールには数多くの便利な機能が付いていますが，これは裏返すと，多量のメモリを使用しているということです。
システムのメモリが不足すると，思わぬトラブルを引き起こす場合があります。十分なメモリを実装しているシステムでも，なるべくコントロール数を倹約するよう心がけてください。例えば，数多いテキストボックスは，表形式コントロールにしましょう。また，1 つのコントロールで，複数の処理を実現することも，メモリの節約には効果的です。

VDS/ASTMAC のトレンドやアラームサマリも多く貼り付けるとメモリを多量に使用します。
「付録 A　パフォーマンスデータ」を参照してください。

2.1.2 I/O アクセス

● I/O アクセスのパフォーマンス見積もり

PLC からデータ収集を行うときには，以下のような最適化を行い，パフォーマンスの向上に努めています。
同一収集周期に重ったデバイスは，連続デバイスとして一括読み込を行います。
I/O ドライバがスキャン最適化ロジック対象であると，連続デバイスアクセス時に以下に説明する最適化処理が行われます。

データ収集の最適化処理

同一の収集周期にあるデータで連続するデバイスは，なるべく一括してデータ収集を行うように I/O 割り付けを行ってください。これにより，通信のパケット数を最小限に抑え，パフォーマンスの向上をはかります。（パフォーマンスデータを見る限り，データ収集のために行われる通信時間は通信回数に大きく依存するためです。）
スキャン実行時に，隣接する I/O アドレスを持つデバイス値の読み出しを，個々のデバイスごとに行うのではなく，装置のデータ転送能力に応じて一括して行います。この際，データアクセスの境界アドレスは，先頭から数えて，装置のデータ転送能力の倍数に対応するアドレスとなります。例えば，データ転送能力が 64 点の装置の場合，境界アドレスは 1，65，129．．．となります。
また，データアクセス境界をまたぐ I/O アドレスが定義されている場合，2 回の I/O アクセスが行われます。

データ取得方式が定周期または変化で同じスキャン周期なら，コントロールオブジェクトの定義の順番，グループに関係なく最適化が行われます。
同じスキャン周期なら同じ I/O アドレスに対して複数のコントロールオブジェクトを定義しても読み込み要求は 1 回のみ出されます。
以下のアクセス方式は最適化の対象外です。要求ごとに通信が行われます。

- 入力系オブジェクトの Refresh メソッド
- 出力系オブジェクトの Cv／Cvi プロパティへの直接データ書き込み
- 出力系オブジェクトの SetCv／SetCvi メソッドによる書き込み
- ブロックデータオブジェクトの ReadIO／WriteIO
- 出力系オブジェクトの汎用インタフェース（SetProperty，SetPropertyies，SetPropertyEx）による書き込み

FA-M3 Ethernet 通信ドライバでの例

1 回のデータ収集に要する時間が 30msec でレジスタ（2Byte）データを読み出す場合の例です。

FA-M3 Ethernet 通信用ドライバは，連続して 498 点のレジスタを一括して読み出すことができます。そのアクセス時間はラダーのスキャン周期やモジュール構成により変わり，30ms～100ms ぐらいです。「付録 A パフォーマンスデータ」を参照ください。

① 連続しない 498 点のレジスタ（2Byte）を読み出すとき（Ex 1,500,1000,…）
14940msec=498 点*30msec

② 連続した 498 点のレジスタ（2Byte）を読み出すとき（Ex 1～498）
30msec=1 パケット*30msec

表　装置（I/O ドライバ別）データ転送能力

	DI	AI（16Bits）	XAI（32Bits）
FA-M3R（Eetehernet）	4096（Write256）	498	249
FA-M3-Ethernet	256	64	32
FA-M3-RS232C	256	64	32
MELSEC-Ethernet	256	256	128
MELSEC-RS232C	256（Write:160）	64	32
SYSMAC（Serial）	480	30	15
（その他）	4096	256	128
EZSocket（MELSEC）	Ezsocket の転送能力はネットワーク種別ごとに異なります。詳細は別表の通りです		

表　EZSocket（ネットワーク種別）データ転送能力

	DI Read	DI Write	AI(16Bits)Read/Write
A 型　RS232C	160	10	64
Q 型　RS232C（115.2kbps 未満）	1024	64	64
Q 型　RS232C（115.2kbps）	4096	256	256
A 型　Ethernet	4096	128	256
Q 型　Ethernet	4096	480	480
MNET，MELSECNET2/10	4096	512	512
CC-LINK	4096	256	256
その他（USB 等）	4096	256	256

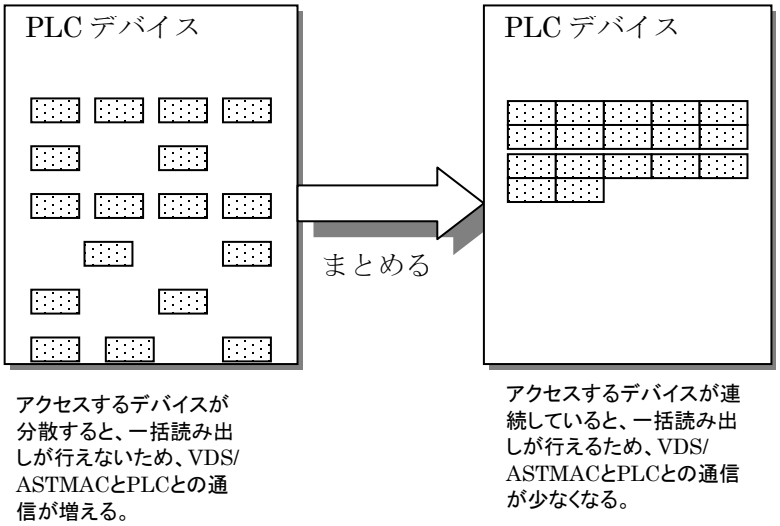
RS 通信のパフォーマンス

RS 通信の場合は通信回数に加えてデータ量にもパフォーマンスは影響されます。例えば FA-M3 RS 通信ドライバを使いワードデバイスを 1 点読み出すのに約 120ms かかりますが，64 点読み出すには倍の約 250ms かかります。「付録 A パフォーマンスデータ」を参照ください。

＜参照資料＞
「VDS/ASTMAC データサーバ解説書（IM34P02D11-01）」「2.5.2.2 I/O アクセスの最適化」

● I/O アクセス設計で注意する点

1. I/O スキャン周期は，デバイスタグオブジェクトで「変化」「定周期」と設定したデバイスがすべて考慮の対象となります。定周期に収集する必要のないオブジェクトは「ワンショット」の設定にし，データサーバもしくはアプリケーションフォームから Refresh メソッドを用い，データ更新を行うことをお勧めします。
2. 周期時間は少ない種類にとどめるようにします。周期時間の種類を多くすると，アクセス回数が増え，正規の収集時間内に更新処理を完了しない可能性があります。
3. デバイス配置
   I/O オブジェクトは，収集するデバイスが連続している場合，I/O 装置との通信において，デバイスの一括読み出しを行うように最適化を実行します。この機能により，収集するデバイスが連続するように配置されていると，非連続のデバイスをアクセスする場合に比べ，I/O 装置との通信効率は飛躍的に向上します。また，連続配置されているデバイスの収集周期がすべて同じであれば，さらに効率的になります。



● ポーリング監視よりイベントの利用

定周期で I/O 装置のデータを監視しているとシステムに負荷がかかります。FA-M3 イーサネット接続か MELSEC イーサネット接続の場合は，PLC のイベント送信サービスを利用することでシステム負荷が軽減されます。

● ブロックデータオブジェクトの“値変化”機能に注意

ブロックデータを“値変化”を使って読み込むタイミングをとると，データサーバは 1 秒周期で I/O アクセスを行っています。2，3 個のブロックデータなら問題ありませんが，多数のブロックデータでこの機能を使うと CPU 負荷があがりデータ収集に影響が及ぶ場合があるので使用を避けてください。

● 非同期要求の注意事項

以下のアクセス手段は最適化処理の対象外ですので注意が必要です。

- 入力系オブジェクトの Refresh メソッド
- 出力系オブジェクトの Cv／Cvi プロパティへの直接データ書き込み
- 出力系オブジェクトの SetCv／SetCvi メソッドによる書き込み
- 出力系オブジェクトの汎用インタフェース（SetProperty，SetPropertyies，SetPropertyEx）による書き込み

これらはデータサーバへの要求が完了して復帰するタイプの要求ではありません。したがって I/O 要求の処理時間を考慮せずにアプリケーションから多量の非同期要求をデータサーバに行うと，要求がキューイングされてアクセスエラーを起こしたり，パフォーマンスの低下やメモリ不足が発生する可能性があります。
完了を確認してから次の要求を出すか，十分な間隔を開けて出すようにしてください。

2.1.3 データサーバの VBA とアプリケーションフォーム

● データサーバの処理時間に注意

データサーバの VBA での処理時間が長くならないように注意してください。
データサーバの VBA は，データサーバが標準として持っているデータ管理機能を補完したり，拡張するために使用することを基本としています。例えば，コントロールオブジェクトのイベントプロシージャを記述したり，標準で用意されているコントロールオブジェクトの機能を拡張する目的で利用します。一方，アプリケーションフォームの VBA には，操業データの管理以外の処理を記述します。操作指示やモニタリングなどの操業画面表示や，データベースアクセス，上位コンピュータとの通信といったさまざまなアプリケーション処理を，複数のフォームに分けて記述します。
データサーバ VBA をアプリケーションフォームで記述すべき処理の代替（例えば汎用リレーショナルデータベースアクセスや上位通信プログラムを記述するなど）として使用することは避けてください。データサーバエンジンの動作が遅延する原因になります。
データサーバの VBA は，画面の作成を行うためのものではありません。したがって，フォームモジュールは使用しないでください。Msgbox などのダイアログも，データサーバエンジンの処理遅延の原因になりますので使用しないでください。

＜主な注意点＞

- MsgBox は使わない！
- ISAM を含めてファイルアクセスはしない！
- AcitveX コンポーネントを組み込んで使用しない！
- ユーザフォームは使用しない！

● データサーバ VBA でのエラー処理

通常エラーとならないところにも OnError 処理を入れて MsgServer オブジェクトを使いどのようなエラーが発生したかをログするようにしてください。
操業時に OnError 処理がないとエラーの個所が特定できずに苦戦することになります。
データサーバは，操業モードでの VBA エラー時に，OnError 処理をしていないとにイベントログに VBA エラーが発生したことをログしますが，関数名や行番号などの情報は付加されません。

● アプリケーションフォーム VBA でのエラー処理

OnError 処理を行っていないとエラーダイアログを出して停止します。OnError 処理を行っていれば動作を続行することができます。OnError 処理を入れてエラーの内容を，メッセージ管理のユーザメッセージを使い記録するようにしてください。操業時に OnError 処理がないとエラーの個所が特定できずに苦戦することになります。
ただし，演算オーバーフローなどは OnError 処理できません。コーディング時や総合テスト時に確認しておいてください。

● プロパティリンクに注意

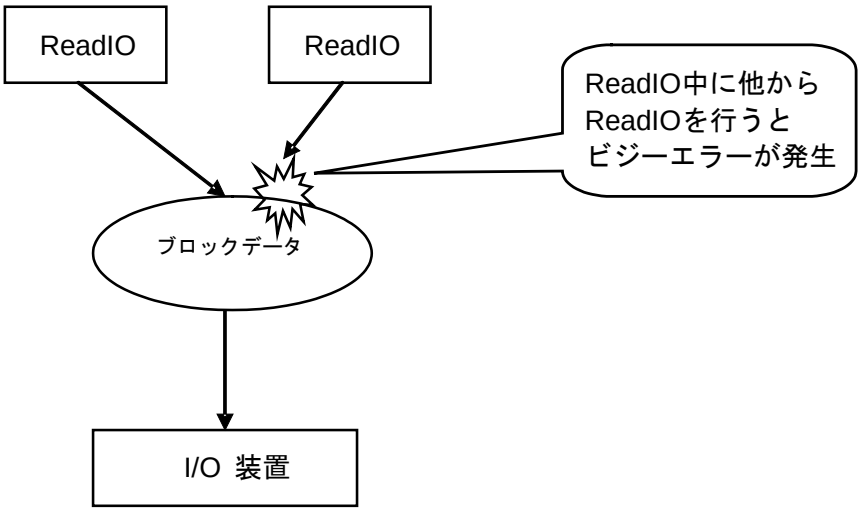
多くのプロパティリンクを張るとアプリケーションフォームのロード時間がかかったり，データ変化時にシステム負荷が高くなる可能性があります。
特に短い周期で変化するデータをリンクすると，再描画が頻繁に発生してシステム負荷を高くするので注意が必要です。

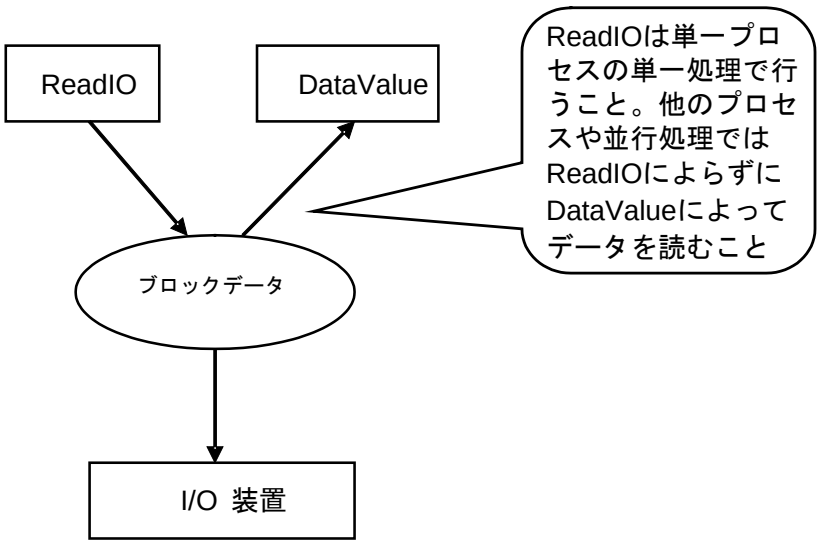
アクティブでないページのプロパティリンクは通常無効になっています。これを有効にする（アクティブモード）と，ページが開いているとデータ変化時の処理が働いてしまいますのでシステム負荷が高くなります。特に必要ない場合，無効にすることをお勧めします。

2.1.4 ブロックデータオブジェクト

● 競合アクセスは避ける

同一のプロセスから完了を待たずに連続して同じブロックデータに I/O 要求を行ったり，複数のプロセスから同時に同じブロックデータに I/O 要求を行ったりすると，ビジーエラーとなります。なお，違うブロックデータに対する I/O 要求に対してはビジーエラーとなることはありません。
1 つのプロセスが定周期で読み込み（ReadIO），そのデータを複数のプロセスが読むように設計してください。ただしデータ量と周期についてはシステムの負荷を考慮して設計してください。





● 書き込み用変数は配列変数を使わずに要素ごとに変数を定義する

R2.05 以降，部分アクセスが可能になりましたが配列変数としては一括書き込みとなります。
一括でなく要素単位に書き込みたい場合は，要素単位に変数を定義してください。
例えば，FA-M3 の D0001 から D0016 を VDS/ASTMAC からの書き込みに使いたい場合は，下記のように構造定義ファイルを作成して WriteIO のパラメータにその該当変数を指定するようにして，部分アクセスを実現してください。

```
Flag1, Integer
Flag2, Integer
Flag3, Integer
・
・
Flag16, Integr
```

「アクセスの例」
YfsDBObj1.WriteIO Flag3

● Microsoft VisualBasic のコンポーネントに注意

Microsoft VisualBasic のコンポーネントのアプリケーションフォームでの動作は保証されておりません。
特に DBGrid や FlexGrid を使うような場面では，VDS/ASTMAC GkitOcx 表形式表示コントロールを使うことを推奨します。
また Microsoft VisualBasic のコンポーネントを動作させるには，Visual Basic の開発環境がインストール されていないと動作しません。VDS/ASTMAC に Microsoft VisualBasic に付属する ActiveX コントロールのファイルとアプリケーションフォームの実行形式ファイルをコピーしてもライセンスがないので動作しません。

2.1.5 その他

● 安易なアンロードは避ける

Grid などの複雑な ActiveX コントロールのイベント内で，使用中のページ，フォームをアンロードすると，通常のイベント処理が完了せず，思わぬトラブルが発生する場合があります。
アンロードが必要な場合には，コマンドボタンなどの標準コントロールのイベントでアンロードするようにします。

VDS/ASTMAC のクロックコントロールとシグナル交信コントロールのイベントプロシージャ内では，使用中のページをアンロードすることは可能です。

● パソコン本体の I/O ポートに注意

- **シリアルポートは受信抜けの可能性がある**

  パソコンのシリアルポートは，パソコンの CPU で直接制御しており CPU へ多くの割り込みを発生します。したがって，ほかの機器との通信の用途で使用する場合や，CPU の負荷が高いときなど，通信中にデータの欠落などが発生する可能性があります。このため市販の拡張カードを使用することを推奨します。
  反対に通信量が多いとシステム負荷をあげる要因となることもあります。

- **パラレルポートはシステム負荷をあげる**

  パソコン本体のパラレルポートについても同様に CPU で直接制御していますので，CPU の負荷が高くなりますので注意が必要です。このような場合は，ネットワークプリンタやプリントサーバを利用することを推奨します。

2.2 システム運用時の障害解析

2.2.1 システム障害解析方法

● システムの応答が遅くなった場合

- タスクマネージャで CPU 負荷をチェックしてください。

CPU 負荷が高い場合は，「プロセス」タブにおいて“CPU”の値が高いプロセスを特定してください。
例えば，アプリケーションフォームとデータサーバの CPU 使用率が高い場合は，アプリケーションフォームがコントロールオブジェクトを頻繁にアクセスしていることが考えられます。
特定のアプリケーションフォームのみの CPU 使用率が高い場合は，そのプログラムの中でルーピングしている可能性があります。

補足

● アプリケーションフォームを特定するには
アプリケーションフォームが複数起動されているとプロセスタブではすべて yfsgrun と表示されるためどのプログラムなのかわからない場合があります。そのときは以下の手順で対応付けてください。
1 「アプリケーション」タブで該当するプログラムを選択して右クリックする。
2 「プロセスの表示」をクリックする。（けしてプロセスの終了をクリックしないこと）
3 「プロセス」タブが，該当する yfsgrun が選択された状態で表示される。

すぐに原因が判明しない場合には，可能ならばデータサーバやアプリケーションフォームをデバッグモードで動かして再発時にデバッグできるようにしてください。
アプリケーションのハングアップが再発する場合は，グラフィックビルダをデバッグモードで起動しておき，デバッグできるようにすることが最善の策です。このときにイミディエイトウィンドウに Debug.Print メソッドでデバッグ文を表示するようにしておけばコードのどこが動作しているかを知ることができます。例えばタイマを使っている場合はタイマイベントプロシージャに Print 文を入れておくと有用です。イミディエイトウィンドウは「表示」メニューの「イミディエイトウィンドウ」を選択すると表示されます。
負荷が高くなった時やハングアップ状態になった場合に，Visual Basic Editor ウィンドウが操作できるなら，F8 キーを押すと動作しているステートメントでブレークすることができます。

- メモリ使用量をチェックしてください

タスクマネージャで「パフォーマンス」タブのメモリ使用量をチェックします。メモリ使用量が以前より極端に増加している場合は，「プロセス」タブの“仮想メモリサイズ”をチェックしてください。極端に増えているプロセスがあるならメモリリークを起こしている可能性があります。
プロセスが特定できたなら，どの機能を使うとメモリリークするかを断定しなければなりません。
市販の ActiveX コントロールを使っている場合は疑ってください。

● FA-M3 との通信でよく起きるエラー

イベントログに記録される FA-M3 通信関連のエラーでオブジェクトビルダの設定の誤りで発生するものを説明します。

ソース名 ドライバ名	番号	エラーコード
YFSIOOBJECT	2004	FA-M3 パソコンリンクコマンドの応答終了コードが返されます。 上位バイト EC1（終了コード） 下位バイト EC2（詳細コード） Ethernet インタフェースモジュール取扱説明書の「3.4 レスポンス終了コード」を参照ください。 0x000052c1：デバイスが存在しない。コントロールオブジェクトの I/O アドレスを見直してください。 0x00000100：CPU 番号指定エラー。コントロールオブジェクトの I/O アドレス内の CPU 番号を見直してください。 0xc101：デバイスの指定エラー。指定したデバイスの範囲が超えているか，コンフィギュレーション設定の範囲外の指定をしています。コントロールオブジェクトの I/O アドレスを見直してください。
	4002	イベント受信ポート Open でのエラーコードが返ります。 同一 VDS/ASTMAC 内でほかの I/O オブジェクトとポート番号が重複している可能性があります。 各 I/O オブジェクトのホスト情報を確認してください。

● アプリケーションフォームがプロジェクトエラーになった場合

「プロジェクトにエラーが発生しました。このプロジェクトの開発者に問い合わせてください。」というダイアログが発生した場合は，次のことを確認してプログラムを修正してください。

グラフィックプロジェクトの実行ファイル（.ygr）を実行しているときに，VBA プログラムに関する実行時エラーが発生した場合に表示されます。実行時エラーは，VBA プログラム内に実行できない操作や演算（例えばオーバーフロー）があったとき発生します。OnError 処理を行っていない場合も発生します。

本エラーが発生した場合には，該当する部分のプログラムを修正する必要があります。

1. グラフィックビルダによってプロジェクトをデバッグ実行させ，実行時エラーを再現させます。このときには具体的にどのようなエラーであるかわかります。
2. その後，VBA プログラムの誤りを修正します。なお，VBA プログラムの実行時エラーについては，デザイン時の VBA のコンパイルでは検知できません。デバッグ実行によって十分なテストを行ってください。

3. FCN/FCJ システム

3.1 レビジョン情報の解析

3.1.1 使用する PC の OS（Windows）とバージョン

FCN/FCJ の開発/保守に必要なツールをインストールする汎用 PC の動作環境は，IM と GS に記述されています。
汎用 PC の OS（Windows）とバージョン，CPU，メモリなどの環境が，動作環境に合致していることを確認します。

● 汎用 PC の動作環境の記述がある IM と GS

- 「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）
- 「NT751FJ ロジックデザイナ」（GS 34P02Q75-01）

汎用 PC の動作環境は，最新の版の IM と GS で確認してください。

● FCN/FCJ 開発/保守ツールや APPF

- リソースコンフィギュレータ
- ロジックデザイナ
- FCN/FCJ シミュレータ
- PAS ポートフォリオ
- そのほかの APPF

3.1.2 FCN/FCJ と開発/保守ツールのレビジョン

FCN/FCJ の基本ソフトウェアのレビジョンと，各開発/保守ツールや APPF の各レビジョンを確認します。

● すべてが最新のレビジョンである場合

FCN/FCJ の基本ソフトウェアと各開発/保守ツールや APPF のレビジョンがすべて最新であれば，その時点でリリースされているすべての機能を使用することができます。

● レビジョンが異なる場合

FCN/FCJ の基本ソフトウェアと各開発/保守ツールや APPF のレビジョンが異なっていると，レビジョンの組み合わせによっては，特定の機能が動作しない，ロジックデザイナのコンパイル時にエラーが発生するなどの問題が発生する可能性があります。

例）

- R1.40 の FCN/FCJ に NPAS_POU を用いたプロジェクトをダウンロードした。
  R1.40 の FCN/FCJ は NPAS_POU をサポートしていないので，PLC エラーが発生します。
- R1.86 のロジックデザイナで新規作成したプロジェクトが，R1.60 の FCN/FC にオンラインダウンロードできない。
  オンラインダウンロードは R1.70 からサポートした機能です。ロジックデザイナが R1.86 であっても，FCN/FCJ が R1.60 であれば，オンラインでのダウンロードはできません。

3.2 LED の状態の解析

3.2.1 電源モジュールの LED の状態

FCN の電源モジュールの正面の上方に［SYS］と［FLD］の 2 個の LED があり，点灯はそれぞれ次の状態を意味しています。

- ［SYS］：5.1V のシステム電源供給中
- ［FLD］：24V アナログフィールド電源供給中

● 5V のシステム電源について

電源モジュールには次の 3 種類があります。

- NFPW441：100～120VAC 入力
- NFPW442：220～240VAC 入力
- NFPW444：24VDC 入力

いずれの電源モジュールでも，上記の入力を 5.1 V DC に変換して供給しており，電源モジュールの［SYS］の LED は，5.1 V DC に変換後の電源の状態を表示しています。

電源モジュールへの入力電源に異常があると［SYS］の LED は消灯します。

また，電源モジュールへの入力電源は正常であっても，電源モジュールの内部の故障などにより，5.1VDC の供給が異常になると［SYS］の LED は消灯します。

● 24V アナログフィールド電源について

24V アナログフィールド電源とは，入出力モジュールにおいて，フィールド信号のインタフェース回路やフィールド機器への電源供給のために必要な 24VDC 電源のことです。［FLD］の LED は，この 24V アナログフィールド電源の供給状態を表示しています。

24V アナログフィールド電源が必要な I/O モジュールを下表に示します。

表　24V アナログフィールド電源が必要な入出力モジュール

形名	機能
NFAI141	16 点電流入力（2 線式で使用する場合）
NFAI143	16 点電流入力
NFAI841	8/8 点電流入出力
NFAI135	8 点電流入力
NFAI835	4/4 点電流入出力
NFAI543	16 点電流出力
NFAP135	8 点パルス入力
NFAB841	8 点電圧入力/ 8 点電流出力

注：　24V アナログフィールド電源が供給されていないと，これらの I/O モジュールは IOP または OOP となり，正常に動作しません。

24V アナログフィールド電源の定格電流出力は 4A ですので，各アナログ入出力モジュールの 24VDC 消費電流の合計は，4A 以下にする必要があります。

消費電流の合計が 4A を上回ると，入出力モジュールは正常に動作しませんが［FLD］の LED は点灯します。

これは消費電流が定格電流を上回っているとはいえ，24V アナログフィールド電源自体は供給されているためです。

● 電源モジュール二重化時の動作

二重化された電源は，突き合わせダイオードにより内部回路に取り込まれますので，どちらか一方の電源モジュールが動作していれば電源は供給されます。

3.2.2 FCN/FCJ の稼働状態 LED の解析

FCN では CPU モジュールの上方，FCJ では正面に，FCN/FCJ の稼働状態を表示する［HRDY］［RDY］［CTRL］の 3 個の LED があります。これらの LED から FCN/FCJ の状態を判断します。
FCN/FCJ の LED の詳細につきましては，次の IM とオンラインヘルプにも記述されていますので，そちらも参照してください。

・ 「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）
「B1.5 FCN/FCJ の動作状態」
・リソースコンフィギュレータのオンラインヘルプ
FCN/FCJ の保守–FCN/FCJ の動作（状態遷移と LED の点灯について）

● FCN/FCJ の動作状態と LED の状態

［HRDY］［RDY］［CTRL］の各 LED は次の 4 通りの状態のいずれかになります。

・点灯
・消灯
・低速点滅：1 秒点灯後 1 秒消灯
・高速点滅：100msec 点灯後 100msec 消灯

［HRDY］［RDY］［CTRL］の状態と，FCN/FCJ の動作状態を以下に示します。LED の状態が以下のパターン以外を示した場合，ベースモジュールのディップスイッチの設定ミスか FCN/FCJ システム異常が発生している可能性があります。

表　FCN/FCJ の動作状態と LED の状態

FCN/FCJ の状態	[HRDY]	[RDY]	[CTRL]	状態の詳細
オンライン状態 （シングル CPU，二重化 CPU の制御側，FCJ）	点灯	点灯	点灯	通常の動作状態です。
オンライン状態 （二重化 CPU の待機側）	点灯	点灯	消灯	通常の動作状態です。
IP アドレス設定状態	高速点滅	低速点滅	消灯	IP アドレスを設定する状態です。
メンテナンス状態	低速点滅	点灯	消灯	FCN/FCJ のコンフィギュレーションなどのメンテナンスが可能な状態です。 OS 機能が動作していて，制御アプリケーションは停止している状態です。
APC 実行中状態（制御側 CPU）	高速点滅	点灯	点灯	二重化 FCN において，制御側 CPU の内容を 待機側 CPU にコピーしている状態です。
APC 実行中状態（待機側 CPU）	高速点滅	高速点滅	消灯	
APC 待ち状態（待機側 CPU）	点灯	高速点滅	消灯	APC の起動要求を待っている状態です。
電源断可能状態/ダウン状態	消灯	消灯	消灯	FCN/FCJ の電源を落としても良い状態。または FCN/FCJ に異常が発生してダウンしている状態。
ブート選択状態	高速点滅	高速点滅	消灯	シャットダウンスイッチの操作で立ち上げモード（IP アドレス・設定・メンテナンス・オンライン）を切り替えることができます。この状態から 3 秒間継続します。
ローディング状態	点灯	低速点滅	消灯	OS をメモリにローディングしている状態です。
ローディング完了状態	点灯	点灯	消灯	FCN/FCJ の立ち上げ中の状態です。

● 通常の動作時の LED の表示

FCN/FCJ が正常に動作している状態では，前述の LED のパターンのオンライン状態になります。
ただし，オンライン状態は OS が動作しているという意味であり，ウォッチドッグタイムエラーや CPU オーバーロードエラー，PLC エラーが発生していても，LED の表示はオンライン状態のままになります。

● LED の表示パターンによる解析

LED の状態が上記のパターン以外を示した場合，以下のことが考えられます。

・ システム的な障害が発生している。
FCN/FCJ の内部に障害が発生し，正常に動作しなくなっている状態です。

・FCN のベースモジュールのディップスイッチの設定が間違っている。
FCN のベースモジュールには，ユニット番号設定と CPU モジュール二重化設定の 2 個のディップスイッチがあります。
これらの設定が間違っていると，FCN が正常に動作しなくなり，LED の表示がおかしくなる場合があります。

例）
- ベースモジュールがユニット 2 の設定であるのに，CPU モジュールを実装した。
- 増設ユニットの設定が，制御ユニットのままであった。
- ベースモジュールがシングル設定であるのに，CPU モジュールを 2 枚実装した。

● 二重化 FCN の LED の表示

二重化 FCN の LED のパターンと稼働状態を，左側 CPU を制御側，右側 CPU を待機側とした場合を例に記述します。（異常が発生していない場合）

表　二重化 FCN の LED の表示

	左側 CPU				右側 CPU			
	[HRDY]	[RDY]	[CTRL]	稼働状態	[HRDY]	[RDY]	[CTRL]	稼働状態
パターン 1	点灯	点灯	点灯	オンライン状態	点灯	点灯	消灯	オンライン状態
パターン 2	点灯	点灯	点灯	オンライン状態	点灯	高速点滅	消灯	APC 待ち状態
パターン 3	点灯	点灯	点灯	オンライン状態	消灯	消灯	消灯	電源断可能状態
パターン 4	点灯	点灯	消灯	オンライン状態	点灯	点灯	点灯	オンライン状態
パターン 5	点灯	高速点滅	消灯	APC 待ち状態	点灯	点灯	点灯	オンライン状態
パターン 6	消灯	消灯	消灯	電源断可能状態	点灯	点灯	点灯	オンライン状態

- パターン 1

左側 CPU が制御権を持つオンライン状態で，右側 CPU が待機側のオンライン状態です。この状態が通常の動作状態になります。

- **パターン 2**

  左側 CPU が制御権を持つオンライン状態で，右側 CPU が APC 待ち状態です。
  待機側（右側）CPU がダウンしたものの，一過性の故障と判断され，CPU に再起動がかかり，APC 待ち状態で立ち上がった状態です。
  右側 CPU の再起動後，APC が実行されていないか，APC が 1 回失敗するとこの状態になります。

- **パターン 3**

  左側 CPU が制御権を持つオンライン状態で，右側が電源断可能状態です。
  右側 CPU ダウンが 1 週間以内に 2 回発生すると，CPU の故障と判断し，再起動は行われません。
  または，右側 CPU のダウン後に再起動できなかった状態です。
  APC が 2 回続けて失敗した場合も，この状態になります。

- **パターン 4**

  本来，待機側であった右側 CPU が制御権を持つオンライン状態で，制御側であった左側 CPU が待機側のオンライン状態になっています。
  制御側であった左側 CPU がダウンし，制御権が右側 CPU に切り替わり，APC が実行された状態です。

- **パターン 5**

  本来，待機側であった右側 CPU が制御権を持つオンライン状態で，制御側であった左側 CPU が APC 待ち状態です。
  制御側であった左側 CPU がダウンし，制御権が右側 CPU に切り替わり，一過性の故障と判断され，左側 CPU に再起動がかかり，APC 待ち状態で立ち上がった状態です。
  左側 CPU の再起動後，APC が実行されていないか，APC が 1 回失敗するとこの状態になります。

- **パターン 6**

  本来，待機側であった右側 CPU が制御権を持つオンライン状態で，制御側であった左側 CPU が電源断可能状態です。
  左側 CPU ダウンが 1 週間以内に 2 回発生すると，CPU の故障と判断し，再起動は行われません。
  または，左側 CPU のダウン後に再起動できなかった状態です。
  APC が 2 回続けて失敗した場合も，この状態になります。

● FCN/FCJ が立ち上がらない障害発生時の解析

1. FCN/FCJ の立ち上げ処理の監視

電源 ON やリセット処理による FCN/FCJ の立ち上げ時，FCN/FCJ は電源断可能状態からオンライン状態に移行します。この時の状態の遷移条件と LED の表示が，前述の IM とオンラインヘルプに記述されています。
FCN/FCJ が立ち上がらない障害が発生した場合，電源の OFF/ON か RESET PB を押し，立ち上げ処理を実行します。
その際，FCN/FCJ の LED を監視し，立ち上げ処理がどこまで実行され，どの段階で中断されるかを確認します。これにより，障害の原因がどこにあるのかを推測することができます。

2. CPU への電源供給の確認

3.2.3 項の FCN/FCJ の通信状態 LED の状態により，FCN/FCJ の CPU に電源が供給されているかの確認を行います。
電源が供給されているにもかかわらず FCN/FCJ が立ち上がらなければ，FCN/FCJ の内部またはシステムカードに障害が発生し，正常に動作しなくなっていると考えられます。
電源モジュールの LED は正常にもかかわらず，CPU に電源が供給されていなければ，電源系統に障害が発生していると考えられます。

3.2.3 FCN/FCJ の通信状態 LED の解析

FCN/FCJ のネットワークインタフェース（LAN ポート）に緑色の LED［LINK］とオレンジ色の LED［ACT］の 2 個の LED があります。
これらの LED の点灯は，それぞれ次の状態を意味しています。

［LINK］：接続正常状態
［ACT］：受信・送信状態

LED［LINK］が点灯しない場合，以下の項目を確認してください。

- FCN/FCJ の電源の状態
  「3.2.1 電源モジュールの LED 状態の解析」を参照してください。
- 接続機器の電源と動作状態
- LAN ケーブル
  ケーブルの種類（ストレート・クロス）が正しいか，断線はしていないか，正しく接続されているか，などを確認してください。

● FCN/FCJ の電源供給状態の確認

FCN/FCJ の LAN ポートの LED［LINK］を用いて，FCN/FCJ の電源供給状態を確認することができます。
接続先の HUB の動作が正常で，かつ FCN/FCJ と HUB が正しく接続されている状態において，FCN/FCJ の CPU に電源が供給されていれば，FCN/FCJ の稼働状態にかかわらず LED［LINK］は点灯します。
接続状態が正常にもかかわらず LED［LINK］が点灯していなければ，各 CPU カードに電源が供給されていないこと意味します。

3.2.4 I/O モジュールの状態表示 LED の解析

I/O モジュールの正面の上方に，I/O モジュールの状態を表示する［STATUS］［ACT］［DX］の 3 個の LED があります。
これらの LED は，それぞれ次の状態を意味しています。

［STATUS］　：ハードウェア正常状態で点灯
［ACT］　：入出力動作中で点灯
［DX］　：未使用のため常時消灯

● ［STATUS］の状態

I/O モジュールの状態が正常であれば，リソースコンフィギュレータでの定義の有無にかかわらず LED［STATUS］は点灯し，故障などにより I/O モジュールの動作が異常にならない限り消灯しません。
これらのことから，LED［STATUS］が消灯している場合は，I/O モジュールに何らかの異常が発生していると判断できます。

● ［ACT］の状態

入出力動作が実行されていれば LED［ACT］は点灯し，I/O の値を更新します。
LED［ACT］が消灯している状態は，入出力動作が実行されていないことを意味しています。（カードによっては外部フィールド電源 24V が供給されていない場合があります）
入出力動作が実行されるのは次の状態ですので，LED［ACT］が消灯している場合は，まずこれらの状態を確認します。（本 LED は一度点灯したら入出力動作が実行されなくても点灯し続けます）

・リソースコンフィギュレータから定義情報がダウンロードされていること。
リソースコンフィギュレータから I/O モジュールに対し，定義情報がダウンロードされていることを確認します。確認方法は，「3.4.1 項システム定義のグローバル変数の解析」を参照してください。
なお，定義情報の中のデバイスラベル名とロジックデザイナで定義したデバイスラベル名が一致している必要があります。

・ロジックデザイナから I/O の切り離しがされていないこと。
I/O を切り離して立ち上げたりすると，FCN の CPU モジュールと I/O モジュールは通信を行わないため，入出力動作は実行されません。

3.2.5 SB バスリピートモジュールの状態表示 LED の解析

SB バスリピートモジュールの正面の上方に，稼働状態と通信状態を表示する［STATUS］［SND］［RCV］の 3 個の LED があります。
これらの LED は，それぞれ次の状態を意味しています。

［STATUS］　：ハードウェア正常状態で点灯
［SND］　：送信中で点灯
［RCV］　：受信中で点灯

LED［STATUS］は，SB バスリピートモジュールの状態が正常であれば点灯します。消灯している場合は，SB バスリピートモジュールに何らかの異常が発生していると判断できます。
LED［SND］と［RCV］は，データの送受信が実行されると点灯します。実際には断続的に送受信が行われるため，LED［SND］と［RCV］は点滅の状態になります。
LED［STATUS］が点灯しているにもかかわらず，LED［SND］と［RCV］が点滅状態にならない場合は，以下の項目を確認してください。

- **ベースモジュールのユニット番号の設定**
  ベースモジュールのユニット番号設定スイッチは，工場出荷時はユニット 1 に設定されています。拡張ユニットに使用するベースモジュールの設定が，ユニット 2 またはユニット 3 となっていることを確認します。

- **SB バスケーブルの状態**
  正常に動作している SB バスリピートモジュールとケーブルを交換し，SB バスケーブルに問題ないことを確認します。

- **SB バスリピートモジュールの実装スロット**
  SB バスリピートモジュールの実装されているスロットを確認します。
  実装可能なスロット位置は，SB バスシングル設定ではスロット 10 のみ，二重化設定ではスロット 9 と 10 です。これ以外のスロットに実装しても動作しません。

- **SB バス二重化の設定**
  SB バスのシングル，二重化の設定は，リソースコンフィギュレータより行います。この設定がシングルであった場合，スロット 9 に実装した SB バスリピートモジュールは動作しません。

3.2.6　シリアル通信モジュールの状態表示 LED の解析

シリアル通信モジュール（NFLR111，NFLR121）の正面の上方に，通信モジュールの状態を表示する［STATUS］［ACT］［DX］の 3 個の LED と，通信状態を表示する
［SND1］［RCV1］［SND2］［RCV2］の 4 個の LED があります。
これらの LED は，それぞれ次の状態を意味しています。

［STATUS］	: ハードウェア正常状態で点灯
［ACT］	: 入出力動作中で点灯
［DX］	: 未使用のため常時消灯
［SND1］	: ポート 1 の送信中で点灯
［RCV1］	: ポート 1 の受信中で点灯
［SND2］	: ポート 2 の送信中で点灯
［RCV2］	: ポート 2 の受信中で点灯

通信モジュールの状態を表示する［STATUS］［ACT］［DX］の動作は，I/O モジュールと同じですので，「3.2.4　I/O モジュールの状態表示 LED の解析」を参照してください。

LED［SND］と［RCV］は，データの送受信が実行されると点灯します。実際には断続的に送受信が行われるため，LED［SND］と［RCV］は点滅の状態になります。
LED［STATUS］と［ACT］が点灯しているにもかかわらず，LED［SND］と［RCV］が点滅状態にならない場合は，以下の項目を確認してください。

- **ケーブルの仕様の確認**

  ケーブルやピンアサインが，仕様に合致しているかを確認します。

- **各ポートの伝送仕様の設定**

  リソースコンフィギュレータより，各ポートの伝送仕様，結線方式や通信方式，伝送速度などの設定を確認します。

- **通信アプリケーションの確認**

  ロジックデザイナでアプリケーションのエラーが発生していないかを確認します。

- **通信内容のトレースの結果**

  3.7.3 項で記述されているリモートシェルツールを用いて通信をトレースし，内容を確認します。

補足

NFLR121 による 2 線式通信においては送信中に［RCV］が点灯する場合があります。

3.2.7 CANopen 通信モジュールの状態表示 LED の解析

CANopen 通信モジュール（NFLC121）の正面の上方に，通信モジュールの状態を表示する［STATUS］［ACT］［DX］の 3 個の LED と，通信状態を表示する［RUN］［COM］の 2 個の LED があります。これらの LED は，それぞれ次の状態を意味しています。

［STATUS］	: ハードウェア正常状態で点灯
［ACT］	: 正常動作中で点灯
［DX］	: 未使用のため常時消灯
［RUN］	: 通信機能正常動作中で点灯，通信機能起動中で点滅
［COM］	: OPERATIONAL で点灯，STOPPED で点滅（200ms ON / 1000ms OFF），PRE-OPERATIONAL で点滅（200ms ON / 200ms OFF）

通信モジュールの状態を表示する［STATUS］［ACT］［DX］の動作は，I/O モジュールと同じですので，「3.2.4　I/O モジュールの状態表示 LED の解析」を参照してください。
［RUN］は，モジュールの状態が正常であれば，リソースコンフィギュレータでの定義の有無にかかわらず点灯し，故障などにより I/O モジュールの動作が異常にならない限り消灯しません。［RUN］が消灯している場合は，I/O モジュールに何らかの異常が発生していると判断できます。
［COM］は，データの送受信が実行されると点灯します。
［STATUS］と［ACT］が点灯しているにもかかわらず，［COM］が点灯状態にならない場合は，以下の項目を確認してください。

- **ケーブル／ターミネータの仕様の確認**

  ケーブル／ターミネータが，仕様に合致しているかを確認します。

- **ポートの伝送仕様の設定**

  CANopen コンフィギュレータで，マスター機器（NFLC121）とスレーブ機器の設定を確認します。

- **システムアラーム，通信エラーの確認**

  CANopen 通信モジュール（NFLC121）から発信されるシステムアラーム，通信アラームが発生していないか確認します。アラーム，エラー，メッセージは，「VDS／ASTMAC メッセージ，アラーム一覧」（TI 34P02K51-01 ）を参照ください。

- **通信アプリケーションの確認**

  ロジックデザイナでアプリケーションのエラーが発生していないかを確認します。

3.2.8 PROFIBUS-DP 通信モジュールの状態表示 LED の解析

PROFIBUS-DP 通信モジュール（NFLP121）の正面の上方に，通信モジュールの状態を表示する［STATUS］［ACT］［DX］の 3 個の LED と，通信状態を表示する［RUN］［COM］の 2 個の LED があります。これらの LED は，それぞれ次の状態を意味しています。

［STATUS］	: ハードウェア正常状態で点灯
［ACT］	: 正常動作中で点灯
［DX］	: 未使用のため常時消灯
［RUN］	: 通信機能正常で点灯，通信機能起動中で点滅
［COM］	: 通信確立状態で点灯(*1)，PROFIBUS-DP コンフィギュレータによるコンフィギュレーションが完了し，バス通信開始指示待ちで周期点滅，PROFIBUS-DP コンフィギュレータによるコンフィギュレーションが未完了で非周期点滅

*1: PROFIBUS-DP コンフィギュレータで定義したスレーブ機器と通信が切断すると消灯となります。

通信モジュールの状態を表示する［STATUS］［ACT］［DX］の動作は，I/O モジュールと同じですので，「3.2.4 I/O モジュールの状態表示 LED の解析」を参照してください。

［RUN］は，モジュールの状態が正常であれば，リソースコンフィギュレータでの定義の有無にかかわらず点灯し，故障などにより I/O モジュールの動作が異常にならない限り消灯しません。［RUN］が消灯している場合は，I/O モジュールに何らかの異常が発生していると判断できます。

［COM］は，PROFIBUS-DP コンフィギュレータで定義した全てのスレーブ機器とのデータの送受信が実行されると点灯します。［STATUS］と［ACT］が点灯しているにもかかわらず，［COM］が点灯状態にならない場合は，以下の項目を確認してください。

- **ケーブル／ターミネータの仕様の確認**

  ケーブル／ターミネータが，仕様に合致しているかを確認します。

- **ポートの伝送仕様の設定**

  PROFIBUS-DP コンフィギュレータで，マスター機器（NFLP121）とスレーブ機器の設定を確認します。

- **システムアラーム，通信エラー，メッセージの確認**

  PROFIBUS-DP 通信モジュール（NFLP121）から発信されるシステムアラーム，通信アラーム，メッセージが発生していないか確認します。アラーム，エラー，メッセージは，「VDS／ASTMAC メッセージ，アラーム一覧」（TI 34P02K51-01 ）を参照ください。

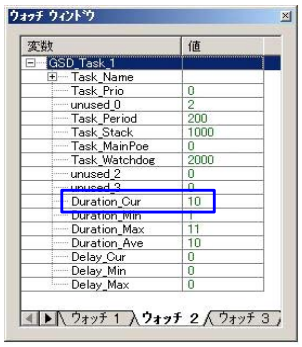
- **通信アプリケーションの確認**

  ロジックデザイナでアプリケーションのエラーが発生していないかを確認します。

3.3 CPU の状態の解析

3.3.1 制御タスクの実行時間

制御タスクの実行時間は，ロジックデザイナのグローバル変数［GSD_Task_**］の Duration_Cur の値から算出されます。
**は 1～16 までで，この数値とタスクの割り付け順が一致しています。



情報収集方法の詳細は，「システム解析ガイド（情報収集編）」（TI 34P02K03-02）「3.4.2 タスクの実行時間」を参照してください。

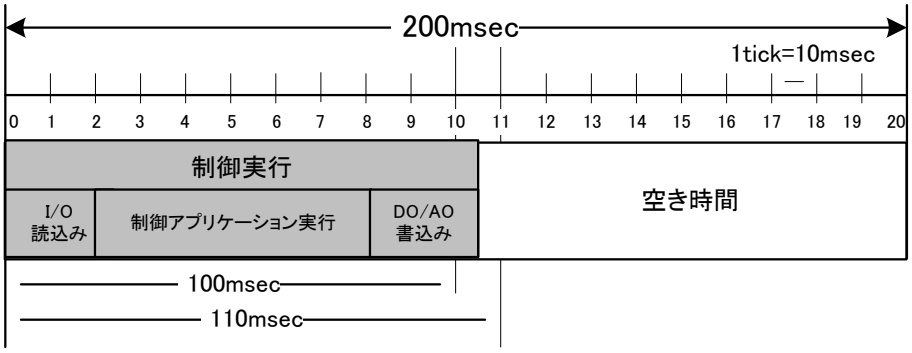
● tick 数について

FCN/FCJ の制御タスクは 10msec 間隔で動作しています。この 10msec の周期を tick と呼びます。
また，制御タスクは tick の回数をカウントしています。これを tick 数と呼びます。
グローバル変数［GSD_Task_**］の Duration_Cur には，現在の制御実行時間が tick 数として表示されています。

● 制御タスクの実行時間の考え方

tick 数と制御タスクの実行時間の考え方を，例を用いて説明します。

例）制御タスク周期=200msec Duration_Cur=10 の場合



1. 制御実行開始時の tick 数を記憶します。
   説明の便宜上，上の図では制御タスク開始時の ticks 数を 0 としています。
2. 10 回目の tick の時点では制御演算は終了していませんでした。
3. 制御演算は 11 回目の tick までに終了したので，11 回目以降の tick では制御演算は行われず，次の制御タスク開始まで空き時間となります。
   この時，制御演算が終了した tick 数を 10 として記憶します。
4. 制御終了時の tick 数と制御開始時の tick 数の差を計算し，制御タスクの実行時間としてグローバル変数［GSD_Task_**］の Duration_Cur に表示します。
   Duration_Cur=10–0=10

- **制御タスク実行時間の考え方**

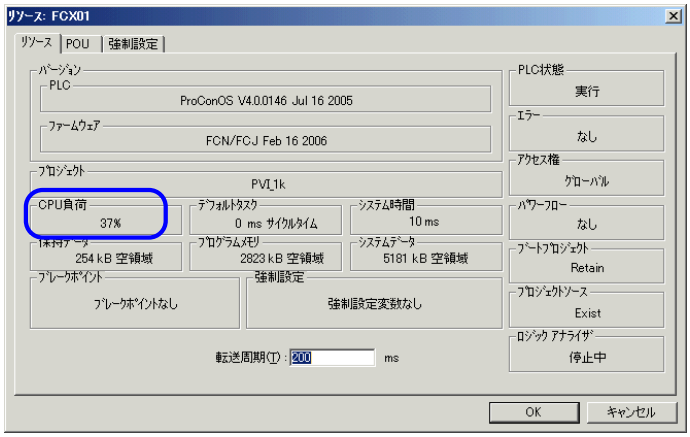
例）

Duration_Cur=10	制御演算が 100msec 以上 110msec 未満で終了。 （上の図の例です。）
Duration_Cur=9	制御演算が 90msec 以上 100msec 未満で終了
Duration_Cur=0	制御演算が 10mse 未満で終了。

制御タスクの処理が制御周期内に終了しなかった場合の動作につきましては，「3.3.3 項ウォッチドッグタイムエラー」を参照してください。

3.3.2 CPU 負荷率の解析

ロジックデザイナのコントロールダイアログの情報ボタンをクリックすると，リソースの情報を表示するダイアログが開きます。ここに CPU 負荷率が表示されます。



情報収集方法の詳細は，「システム解析ガイド（情報収集編）」（TI 34P02K03-02）「3.4.1　CPU 負荷率」を参照してください。

● CPU 負荷率の計算方法と更新周期

FCN/FCJ は 10msec の tick ごとに制御アプリケーションのタスクの動作状態をチェックしています。
チェック時に制御タスクが動作していれば，そのタスクが CPU を使用していると判断し，タスクが動作していた回数をカウントします。
このチェックを 1024 回（1024×10msec=10.24 秒）継続して行い，CPU 負荷率を計算します。

例えば，1024 回のチェック時に制御タスクの動作カウント数が 100 回であれば，

CPU 負荷率=100 回/1024 回 x100%=9.765・・・・≒ 10%

となります。

● CPU 負荷率の考え方

例）タスク周期：200msec，CPU 負荷率：37%の場合

CPU 負荷率の 37%は 1024 回のチェック時に制御タスクが動作していた割合であり，制御タスクが毎周期 74msec（=200msec×37%）で実行されているわけではありません。

・制御タスクの実行回数

10.24 秒のチェック間隔における 200msec 周期の制御タスク実行回数は 51 回になります。

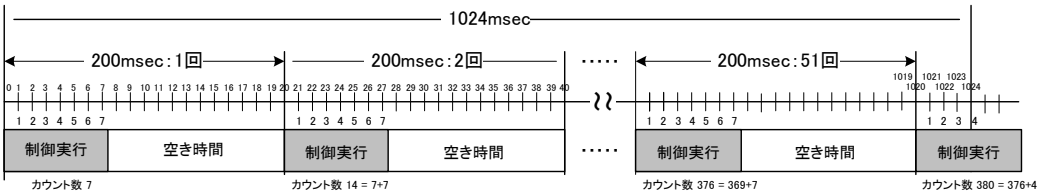
10.24sec/200msec=51.2=51 回

・制御タスクの動作カウント回数

CPU 負荷率 37%から，1024 回のチェックのうち，制御タスクは 380 回動作していたことになります。

1024×37%=378.88=380 回

・CPU 負荷率の考え方



10.24 秒のチェック間隔内に CPU の負荷にバラつきがなく，ほぼ一定であったと仮定します。
この仮定において，制御タスクが 51 回実行された中で，制御タスクの動作回数 380 回を平均的に振り分けます。
その結果，動作カウント数が 7 であったタスク周期が 28 回，カウント数が 8 であったタスク周期が 23 回であったと考えることができます。

カウント数 7×28 回=196 回
カウント数 8×23 回=184 回
制御タスク実行回数　　= 51 回= 28 回+ 23 回
制御タスク動作カウント回数 =380 回=196 回+184 回

これらをまとめると，タスク周期 200msec において CPU 負荷率 37%とは，10.24 秒の CPU 負荷率計算周期の間に制御タスクが 51 回起動され，そのうち 28 回は制御演算が 70msec 以上 80msec 未満で終了し，23 回は 80msec 以上 90msec 未満で終了したことになります。

重 要

この結果は，CPU の負荷にバラつきがなく，毎周期の処理時間がほぼ同じであると仮定した計算であり，実際の状態とは異なる場合があります。
上述の例において，51 回の実行された制御タスクの中で，49 回はカウント数が 7 で動作していても，突発的な CPU 負荷増大により，カウント数が 18 となったタスク周期が 2 回発生すれば，CPU 負荷率は 37%になります。
CPU の負荷のバラつきを確認するには，グローバル変数［GSD_Task_**］をロジックデザイナのウォッチウィンドウに割り付け，Duration_Cur の値を目視で監視します。
または Duration_Cur の値を UINT 型の変数に代入し，この変数をロジックアナライザで解析します。

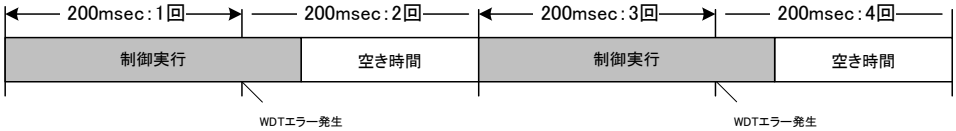
3.3.3 ウォッチドックタイムエラー

ロジックデザイナのタスク設定ダイアログに，ウォッチドッグ時間（以下，WDT と記述）の設定個所があります。
ここで WDT を設定すれば，制御の実行が WDT 以内に終了しなかった場合に WDT エラーが発生し，ログファイルに記録されます。

例）
200*/**/****:**:**,・・略・・,Msg=Msg=Exception(Watchdog) in Task0,

● WDT エラー発生動作

例）制御タスク周期=200msec　WDT=200msec の場合の動作



この図では，制御タスクの処理に約 250msec かかっており，1 回の制御周期内に処理が終了していません。1 回目の制御タスクの処理が 2 回目のタスク周期で終了しています。

1 回目の制御タスク	制御タスク周期内に制御演算が終了していないため，タスク終了時に WDT エラーが発生します。
2 回目の制御タスク	1 回目の制御タスクで処理しきれなかった残りの演算を行います。この演算は約 50msec で終了しますが，残りの約 150 msec で次の制御演算を行わず，空き時間となります。
3 回目の制御タスク	次の制御が実行されますが，制御タスク周期内に演算が終了していないため，タスク終了時に WDT エラーが発生します。
4 回目の制御タスク	2 回目と同様に約 50msec で演算を終了し，残りの 150 msec は空き時間となります。

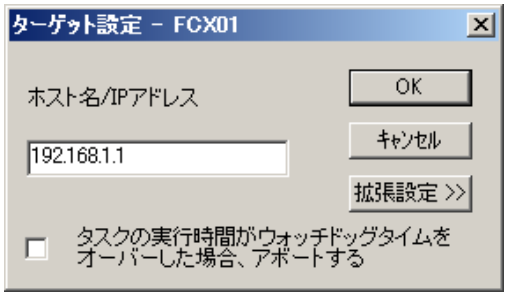
この例では，200msec の制御タスク周期内に制御実行が終了しないため WDT エラーが発生し，本来ならば 800msec の間に 4 回実行されるはずの制御実行が 2 回しか実行されず，実質的なタスク周期は設定の倍の 400msec となります。

● WDT の設定について

- WDT の設定は，10msec 単位で任意の値が設定できます。
- 制御タスク周期＝WDT,
  制御タスク周期＞WDT,
  制御タスク周期＜WDT
  のいずれの設定でも問題ありません。
- 設定を 0 msec とすれば WDT エラーの監視を行いません。

● 注意事項

ロジックデザイナの TargetSetting のダイアログに『タスクの実行時間がウォッチドッグタイムをオーバーした場合，アボートする』のチェックボックスがあります。このチェックボックスから，WDT エラー発生時の制御タスクの動作を指定します。



チェックあり： WDT エラー発生で制御タスクが停止します。
チェックなし： WDT エラー発生がログファイルには記録されますが，制御タスクは停止しません。

デフォルトの設定は『チェックなし』となっています。設定を『チェックあり』に変更すると，WDT エラーにより制御タスクが停止しますので，この設定を変更する場合は，十分注意してください。
また，このチェックを『なし』として WDT を 0 以外に設定すれば，タスクを停止せずに WDT エラー発生の監視をすることができます。

補足

二重化 CPU の場合は，設定を『チェック無し』にしてください。

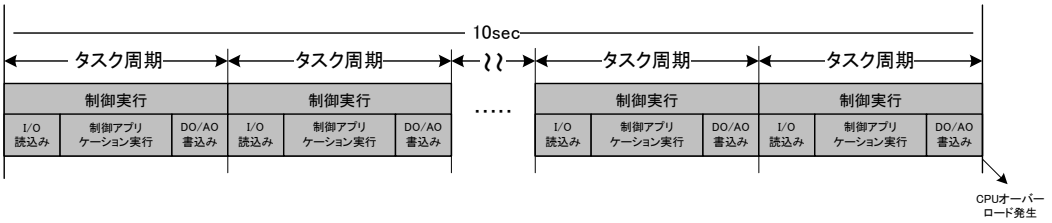
3.3.4 CPU オーバーロード

CPU の負荷が増大し，CPU の空き時間がなくなった状態が 10 秒間継続すると CPU オーバーロードエラーが発生し，ログファイルに記録されます。
CPU オーバーロードエラーが発生する状況は，次の 2 通りが考えられます。
例）

200*/**/****:**:** , ・・略・・, Msg=Exception(CPU Overload) in Task0,

● パターン 1：制御演算がタスク周期内に終了する場合

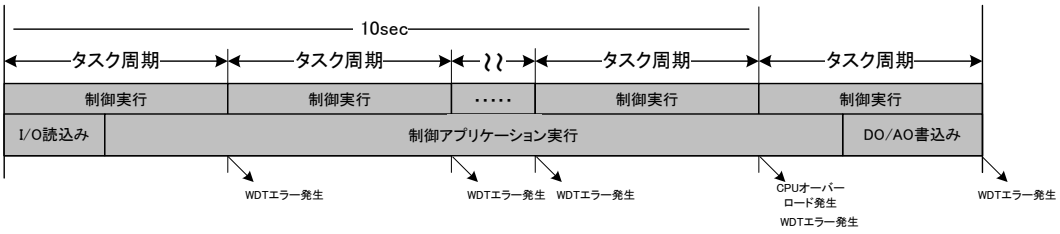
下の図では，制御演算はタスク周期内で終了しているものの，周期内ぎりぎりであり，CPU の空き時間が 0 となっています。
この状態が 10 秒間継続すると，CPU オーバーロードエラーが発生します。
また，制御タスク周期＝WDT の設定をしてあっても，制御演算がタスク周期内で終了しているため，WDT エラーは発生しません。



この場合，制御演算の処理量に対してタスクが短いため，恒常的に CPU 負荷が高くなっていることが考えられます。

● パターン 2：制御演算がタスク周期内に終了していない場合

下の図では，1 回の制御演算の処理に時間がかかり，1 回処理が 10 秒以上継続して実行されています。制御演算の実行が 10 秒継続したタイミングで，CPU オーバーロードエラーが発生します。
また，下図の例では WDT＝タスク周期の設定してあるため，各タスク周期の終了時に WDT エラーが発生しています。



この場合，下記の様な原因で制御演算が終了しなくなったと考えられます。

- **制御アプリケーションの繰り返し処理が終わらなくなった。**

例）

ST 言語の FOR 文の処理や，ジャンプとラベルの処理の間違いにより，アプリケーションが無限ループの状態になったため，1 回の制御演算の処理が 10 秒以上になってしまった。

- **制御タスクより優先順位の高いタスクの処理が継続して行われた。**

例）

FCN/FCJ において，Ethernet 通信に対する CPU の処理は，制御タスクより優先順位が高くなっています。
何らかの理由で Ethernet 上の機器が FCN/FCJ に高負荷となる通信を続けて行うと，FCN/FCJ はその処理を優先させるため，制御タスクは待ち状態になります。その状態が 10 秒以上継続した。

注： よくある例としてブロードキャスト通信があげられます。

● CPU オーバーロードエラー発生時の動作

CPU オーバーロードエラーが発生した状態では，CPU の空き時間が 0 となっています。そのため，通常は空き時間で行われる以下の処理が実行できなくなります。これらはパターン 1 とパターン 2 で共通の動作となります。

- 原因となったタスクより低いプライオリティのタスク動作
- VDS/ASTMAC のデータサーバとの通信
- FCN/FCJ 間通信
- Modbus 通信など，Ethernet 通信またはシリアル通信を用いた各機器間の通信
- InfoWell 機能
- ユーザ作成の Java アプリケーション

● パターン 1 とパターン 2 の動作の違い

- **パターン 1**

CPU の空き時間が 0 となっていますが，演算処理は各タスク周期ごとに終了しています。
そのため，アナログ・デジタルの入出力は各タスク周期ごとに行われます。
また，WDT=制御周期の設定をしてても，WDT エラーにはなりません。

- **パターン 2**

演算処理が 1 回のタスク周期内で終了していないため，アナログ・デジタルの入出力処理が各タスク周期で行われない状態が 10 秒以上継続されています。
これは I/O モジュールが CPU の異常を検出する『ラインアクセス喪失時間：4 秒』を越えていますので，I/O モジュールは出力フォールバックの動作を行います。
また，WDT=タスク周期の設定をしていれば，タスク周期ごとに WDT エラーが発生します。

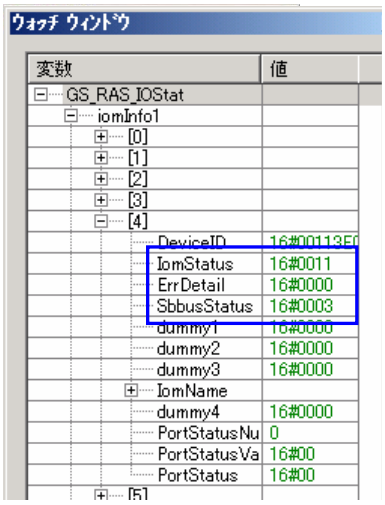
3.4 ロジックデザイナの情報

3.4.1 システム定義のグローバル変数の情報

3.3.1 項で説明した［GSD_Task_**］以外にも，システムの状態を表示するグローバル変数があります。

● GS_RAS_IOStat

このグローバル変数は，実装されている I/O モジュールの状態を表示します。
下の図では，制御ユニット（unit1）の 5 スロットに実装されている I/O モジュールの状態を表示しています。これらの情報の中から，四角枠で囲われている IomStatus，ErrDetail，SbbusStatus について説明します。



情報収集方法の詳細は，「システム解析ガイド（情報収集編）」（TI 34P02K03-02）「3.4.3 システム定義のグローバル変数の情報」を参照してください。

・ IomStatus

IomStatus は WORD 型データであり，I/O モジュールの状態を表示します。
各ビットの意味は以下のとおりです。
これらを組み合わせると，表示が 16#0011（定義ありで正常動作中），または 16#0013（定義ありでダウンロード中）が正常状態になります。

表　IomStatus

値	説明
16#0001	正常動作中 I/O モジュールでシステムアラームが発生していない状態です。 システムアラームが発生する条件は外部供給電源有/無など，IOM の種類によって異なります。
16#0002	ダウンロード中 リソースコンフィギュレータから定義情報をダウンロード中の状態です。
16#0010	定義あり リソースコンフィギュレータから定義情報がダウンロードされていることを意味します。
16#0100	スタンバイ状態 NFLF111 モジュールの二重化時のみで表示されます。

▪ ErrDetail

ErrDetail は WORD 型データであり，I/O モジュールのエラー情報を表示します。

表　ErrDetail

値	説明
16#0000	正常状態 未定義の時もこの値が表示されます。
16#0001	異常状態 以下のいずれにも該当しない異常状態です。
16#0002	コンフィギュレーションエラー 正常運転している IOM を同一形名のものと交換し，交換前後で IOM のコンフィギュレーションが異なる場合，この値が表示されます。 「自動ロードを有効にする」オプションが有効であれば，この状態をトリガとして自動ロードが行われ，異常は解消されます。
16#0003	IOM 形名エラー 正常運転している IOM を別の形名のものと交換した場合，この値が表示されます。
16#0004	IOM にファームウェアがロードされていない
16#0005	IOM のファームウェアのレビジョン不一致

▪ SbbusStatus

SbbusStatus は WORD 型データであり，I/O モジュールに接続している SB バスの状態を表示します。各ビットの意味は以下のとおりです。

表　SbbusStatus

値	説明
16#0001	SB バス A 系正常
16#0002	SB バス B 系正常 SB バスシングル設定時は B 系だけが使用されますので，この値が表示されます。
16#0003	SB バス A 系，B 系とも正常動作（二重化設定時）

SbbusStatus の値が，上記の正常値以外の表示をしていれば，SB バスのどこかに異常があると判断します。
この SB バスには，ベースモジュール内部の SB バス，SB バスリピートモジュール，SB バスケーブルだけでなく，I/O モジュール内部の SB バスも含まれます。
そのため，I/O モジュールの単体故障であっても，該当モジュールの SbbusStatus の値が異常になる場合があります。
SbbusStatus の値が異常となった場合，次にそれが SB バスの異常か，I/O の単体故障かを判断します。
それには，同一ベースモジュールの実装されているほかの I/O モジュールの SbbusStatus の値を確認します。
同一ベースモジュール上のすべての I/O モジュールの SbbusStatus が異常であれば，ベースモジュール内 SB バス，SB バスリピートモジュール，SB バスケーブルのいずれかに異常があると判断します。
同一ベースモジュール上のほかの I/O モジュールの SbbusStatus が正常であれば，該当 I/O モジュールの単体故障と判断します。

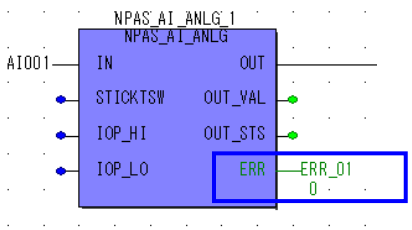
● GS_RAS_Current

GS_RAS_Current は，現状の RAS 情報を表示します。
この中には，システムカードのシリアル No.やレビジョン情報，CPU の動作状態，ダウン情報などが含まれています。これらの情報は，リソースコンフィギュレータ上の RAS 情報に一括して表示されます。
表示情報の詳細につきましては，「3.5.3 RAS 情報の解析」を参照してください。

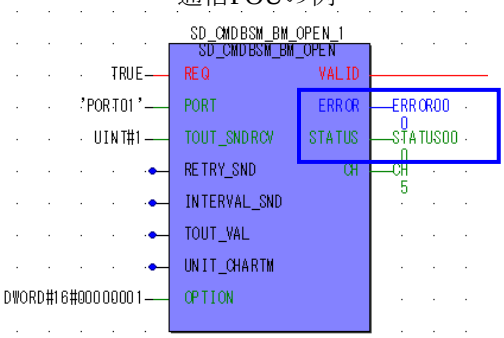
3.4.2 POU のエラー情報

NPAS_POU の入出力データ処理 POU と各通信 POU には，エラー発生とエラー状態を表示する出力端子があります。
この端子に変数を接続し，ロジックデザイナをデバッグモードにすると値が表示されます。
エラーコードの詳細は，該当 POU のオンラインヘルプに記述されています。

入出力POUの例



通信POUの例



特に通信が正常に動作しない場合，通信 POU に表示されるエラーコードは，原因解明のための有効な情報になります。
通信エラー発生時の情報収集のため，通信 POU の ERROR 端子と STATUS 端子には，あらかじめ変数を割り付けておくことを推奨します。

3.4.3 ロジックデザイナのログ情報の解析

ロジックデザイナのログファイルは下記のフォルダに格納されています。

<ロジックデザイナのインストールパス>¥LogicDesigner¥Mwt¥Log

標準のインストールパスの場合，以下のフォルダになります。

C:¥YOKOGAWA¥FCN-FCJ¥LogicDesigner¥Mwt¥Log

このログファイルに，ロジックデザイナでプロジェクトをメイク，またはコンパイルを行った日時が，以下の内容で記録されます。（R1.70 以降のみ）

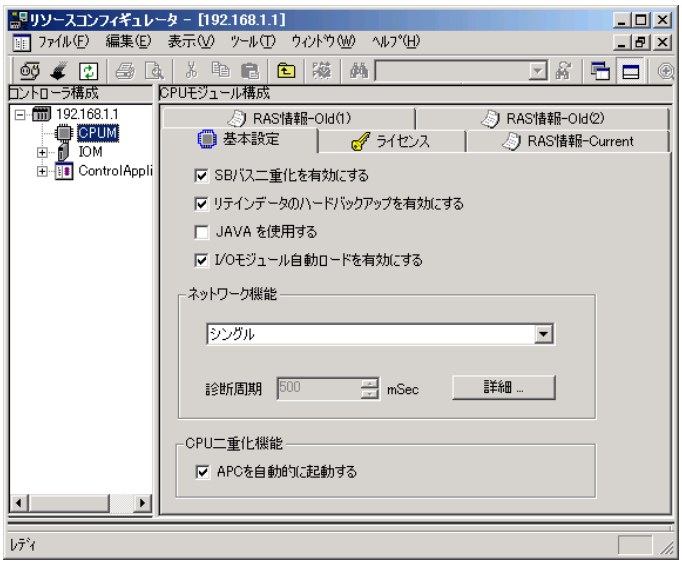
例)

20**/**/** **:**:** TRACE,0x000f, C:¥BuildEnv¥VC¥Src¥LogicDesigner¥SdcsWriteGenInfo¥CSdcsEngLog.h:106,"---- Writing generational information FCX01----"

3.5 リソースコンフィギュレータの情報の解析

3.5.1 CPU モジュールの基本設定

「基本設定」には，FCN/FCJ の基本設定の情報が表示されます。



● SB バス二重化を有効にする

ここにチェックがあると，SB バスが二重化されます。
SB バス二重化については，「3.2.5　SB バスリピートモジュールの状態表示 LED の解析」を参照してください。

● リテインデータのハードバックアップを有効にする

ここにチェックがあると，リテインデータが CPU の SRAM（不揮発性メモリ）上に保持されます。
リテインデータに関する障害が発生した場合，この設定を確認します。

● JAVA を使用する

シングル FCN，または FCJ で JAVA プログラムを使用する場合にチェックします。
システムカードに JAVA 機能実行ライセンスが登録されていても，このチェックが外れていると，JAVA 機能は動作しません。
JAVA プログラムが動作しない場合，この設定を確認します。

● I/O モジュール自動ロードを有効にする

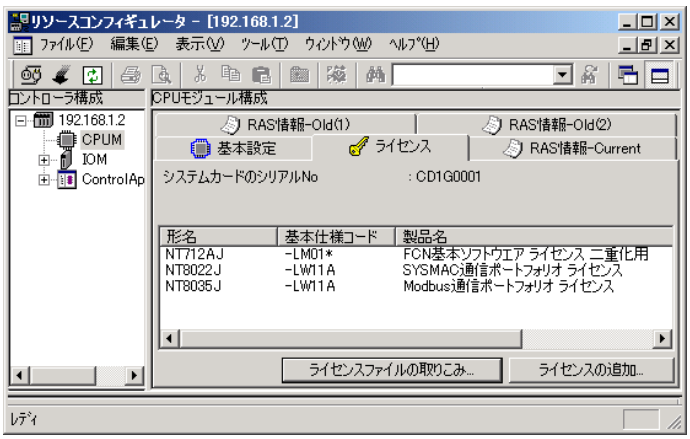
ここにチェックがあると，I/O モジュール交換後，リソースコンフィギュレータで定義されていた設定情報が自動的に再ロードされます。
I/O モジュールが交換後に動作しない場合，この設定を確認します。
I/O モジュールに関する詳細は「3.2.4 I/O モジュールの状態表示 LED の解析」，「3.4.1 システム定義のグローバル変数の情報の解析」を参照してください。

● APC を自動的に起動する

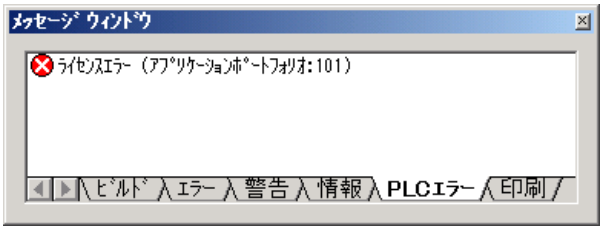
ここにチェックがあると，二重化 FCN の自動 APC の機能が有効になります。
二重化 FCN に関する障害発生した場合，この設定を確認します。

3.5.2 システムカードのライセンス情報

「ライセンス」には，システムカードに登録されているライセンスの一覧が表示されます。



ここに登録されていていない APPF を含むプロジェクトをロジックデザイナからダウンロードすると，ロジックデザイナのメッセージウィンドウにライセンスエラーが表示されます。



表示されたエラーを右クリックし，メッセージのヘルプを選択すると，コントローラエラーメッセージヘルプが表示されます。このヘルプに，エラーとして表示されている APPF の番号の一覧が記載されています。

表　アプリケーションポートフォリオ一覧（1/2）

番号	形名	アプリケーションポートフォリオ
505	NT730AJ	フィールドネットワークモジュール二重化ライセンス
101	NT8001J	PAS ポートフォリオライセンス
102	NT8002J	SAMA ポートフォリオライセンス
105	NT8005J	汎用ポートフォリオ A ライセンス
110	NT8010J	Webmetry 基本ライブラリポートフォリオライセンス
112	NT8012J	InfoWell Web アプリケーションポートフォリオライセンス
113	NT8013J	InfoWell E-mail アプリケーションポートフォリオライセンス
114	NT8014J	InfoWell グラフィックポートフォリオライセンス
115	NT8015J	Graphic アプリケーションポートフォリオライセンス
116	NT8016J	ロギングポートフォリオライセンス
117	NT8017J	InfoWell ライセンス
120	NT8020J	FA-M3 通信ポートフォリオライセンス
121	NT8021J	MELSEC 通信ポートフォリオライセンス
122	NT8022J	SYSMAC 通信ポートフォリオライセンス
130	NT8030J	電力モニタ通信ポートフォリオライセンス
131	NT8031J	温調計通信ポートフォリオライセンス
132	NT8032J	JUXTA VJ 通信ポートフォリオライセンス
133	NT8033J	PR488 通信ポートフォリオライセンス
135	NT8035J	Modbus 通信ポートフォリオライセンス
136	NT8036J	DNP3 通信ポートフォリオライセンス
140	NT8040J	時刻同期サーバポートフォリオライセンス

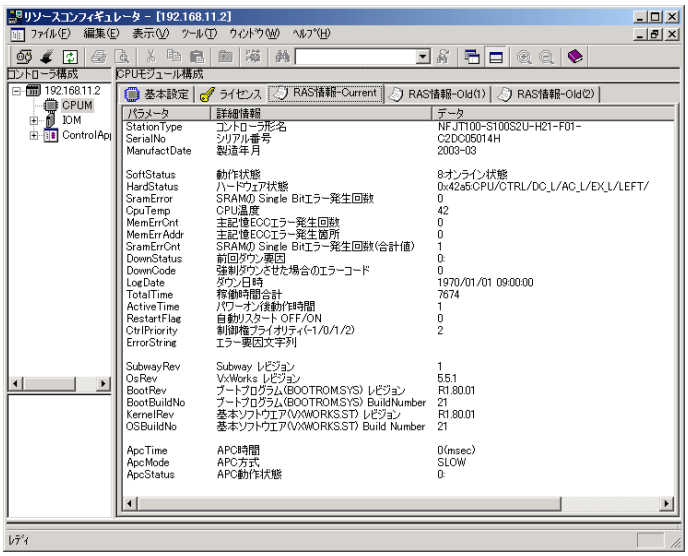
表　アプリケーションポートフォリオ一覧（2/2）

番号	形名	アプリケーションポートフォリオ
201	NT8101J	ボイラ制御ポートフォリオライセンス
202	NT8102J	ボイラ補機制御ポートフォリオライセンス
203	NT8103J	スケジュール発停ポートフォリオライセンス
205	NT8105J	ガス流量演算ポートフォリオライセンス
700	NT8600J	InfoEnergy デマンド監視ポートフォリオライセンス
900	NT8800J	eCUBE コンフィギュレータライセンス
901	NT8801J	eCUBE ユニットポートフォリオライセンス（極小型）
902	NT8802J	eCUBE ユニットポートフォリオライセンス（小型）
903	NT8803J	eCUBE ユニットポートフォリオライセンス（中型）
904	NT8804J	eCUBE ユニットポートフォリオライセンス（標準型）
909	NT8809J	eCUBE ユニットポートフォリオライセンス（カスタム型）
911	NT8811J	エコノパイロット ライセンス 1 系統用（削減リミット 150MWh）
912	NT8811J	エコノパイロット ライセンス 1 系統用（削減リミット 300MWh）
913	NT8811J	エコノパイロット ライセンス 1 系統用（削減リミット 500MWh）
914	NT8811J	エコノパイロット ライセンス 1 系統用（削減リミット 700MWh）
915	NT8811J	エコノパイロット ライセンス 1 系統用（削減リミット 無制限）
919	NT8811J	エコノパイロット ライセンス 1 系統用（OEM）
921	NT8812J	エコノパイロット ライセンス 2 系統用 1 系統目（削減リミット 150MWh）
922	NT8812J	エコノパイロット ライセンス 2 系統用 1 系統目（削減リミット 300MWh）
923	NT8812J	エコノパイロット ライセンス 2 系統用 1 系統目（削減リミット 500MWh）
924	NT8812J	エコノパイロット ライセンス 2 系統用 1 系統目（削減リミット 700MWh）
925	NT8812J	エコノパイロット ライセンス 2 系統用 1 系統目（削減リミット 無制限）
929	NT8812J	エコノパイロット ライセンス 2 系統用 1 系統目（OEM）
931	NT8813J	エコノパイロット ライセンス 2 系統用 2 系統目（削減リミット 150MWh）
932	NT8813J	エコノパイロット ライセンス 2 系統用 2 系統目（削減リミット 300MWh）
933	NT8813J	エコノパイロット ライセンス 2 系統用 2 系統目（削減リミット 500MWh）
934	NT8813J	エコノパイロット ライセンス 2 系統用 2 系統目（削減リミット 700MWh）
935	NT8813J	エコノパイロット ライセンス 2 系統用 2 系統目（削減リミット 無制限）
939	NT8813J	エコノパイロット ライセンス 2 系統用 2 系統目（OEM）
951	NT8851J	eCUBE 帳票パッケージ UNIT ライセンス
953	NT8853J	eCUBE 現場監視パッケージ UNIT ライセンス
955	NT8855J	eCUBE イベントデータ表示パッケージ UNIT ライセンス
957	NT8857J	eCUBE 警報パッケージ UNIT ライセンス
959	NT8859J	eCUBE ファイル変換パッケージ UNIT ライセンス
962	NT8862J	eCUBE イベント接点機能 UNIT ライセンス
963	NT8862J	eCUBE Reserved Option License
970	NT8870J	eCUBE aqua RO 診断ポートフォリオライセンス
971	NT8861J	eCUBE CSV 変換機能 UNIT ライセンス
979	NT8885J	InfoEnergy ソフトウェアカスタマイズ用ライセンス
980	NT8880J	InfoEnergy ソフトウェアライセンス
981	NT8885J	InfoEnergy ソフトウェアライセンス 横河製電力モニタ接続：62 台
982	NT8885J	InfoEnergy ソフトウェアライセンス 横河製電力モニタ接続：124 台
983	NT8885J	InfoEnergy ソフトウェアライセンス 直アナログ入力：6 点（FCJ にのみ適用可）
984	NT8885J	InfoEnergy ソフトウェアライセンス 直アナログ入力：16 点（FCN にのみ適用可）
985	NT8885J	InfoEnergy ソフトウェアライセンス 直アナログ入力：32 点（FCN にのみ適用可）
986	NT8885J	InfoEnergy ソフトウェアライセンス 直デジタル入力：16 点（FCJ にのみ適用可）
987	NT8885J	InfoEnergy ソフトウェアライセンス 直デジタル入力：64 点（FCN にのみ適用可）
988	NT8885J	InfoEnergy ソフトウェアライセンス 直デジタル入力：128 点（FCN にのみ適用可）

このヘルプの内容から，上の図の「アプリケーションポートフォリオ：101」は PAS ポートフォリオのライセンが登録されていないために発生していることがわかります。

3.5.3 RAS 情報の解析

「RAS 情報–Current」には，ロジックデザイナのシステム定義のグローバル変数「GS_RAS_Current」の情報が表示されます。



● ハードウェアに関する情報

CPU モジュールのシリアル番号，製造年月，動作状態，CPU 温度，基本ソフトウェアのレビジョンとビルドナンバー，APC の情報などが表示されます。
特に基本ソフトウェアのレビジョンとビルドナンバーは，現象の解析を行う上で必須の情報です。

● エラーやダウン要因に関する情報

表　RAS 情報のエラーやダウン要因に関する情報

パラメータ	詳細説明
SramError	電源 ON から今までの SRAM のシングルビットエラー発生回数。
SramErrCnt	製造から今までの SRAM のシングルビットエラー発生回数の合計値。
DownStatus	前回のダウンステータスがビットフラグで示される。
DownCode	DownStatus がソフトウェアによる強制シャットダウンの場合，エラーコードがセットされる。
LogDate	シャットダウンされた日付と時刻。
ErrorString	エラー要因文字列。

これらの情報の中で，LogDate と DownCode より，FCN/FCJ がダウンした日時と要因を知ることができます。
これらの情報の詳細は，リソースコンフィギュレータのオンラインヘルプ–［FCN/FCJ 設定方法］–［RAS 情報を見る］に記述されています。

● RAS 情報 – Current, RAS 情報- Old(1), -Old(2)について

Current：現状の RAS 情報を表示します。
FCN/FCJ は動作中であるため，ダウン日時には「1970/01/01 09:00:00」が表示されます。
Old(1)：1 回前に FCJ が起動した時の RAS 情報を表示します。
Old(2)：2 回前に FCJ が起動した時の RAS 情報を表示します。

これらの情報は，次のタイミングで Current → Old(1) → Old(2)へ移されます。

- FCN/FCJ のシャットダウン処理により電源断可能状態に移行したタイミング。
  シャットダウン処理により，DownStatus と DownCode，LogDate が記録されます。
- シャットダウン処理無しで電源断となり，FCN/FCJ が立ち上がったタイミング。
  これは停電により FCN/FCJ がいきなり電源断となり，復電したような場合です。シャットダウン処理が行われないため，DownStatus と DownCode には 0 が，LogDate には FCN/FCJ が立ち上がった日時が記録されます。
- シャットダウン処理無しで FCN/FCJ のリブート処理が行われたタイミング。
  これは FCN/FCJ 稼働中に RESET ボタンを ON にされた，FCN/FCJ のメンテナンスページからリブートされた，などの場合です。
  DownStatus には RESET ボタン ON が記録されますが，LogDate には FCN/FCJ が立ち上がった日時が記録されます。

3.5.4 プロジェクトの情報

プロジェクトの情報には，現在動作中のプロジェクトの名前と日付情報だけでなく，ブートプロジェクトと圧縮プロジェクト（プロジェクトソース）の名前と日付情報も表示されます。これらの日付情報は，以下の日時を表示しています。

- **プロジェクト日付**

  現在動作中のプロジェクトがメイク/コンパイルされた日時が表示されます。
  プロジェクトがダウンロードされた日時ではありません。

- **ブートプロジェクト日付**

  ブートプロジェクトがメイク/コンパイルされた日時が表示されます。
  プロジェクトがダウンロードされた日時ではありません。

- **圧縮プロジェクト日付**

  プロジェクトが圧縮された日時が表示されます。
  プロジェクトソースはダウンロード時に圧縮されますので，圧縮された日時とダウンロードされた日時は一致します。

重 要

上記の情報は，プロジェクト，ブートプロジェクト，プロジェクトソースのダウンロードを行ってから 15 分間は更新されません。
また，ロジックデザイナのリソースコントロールダイアログ表示中や，デバッグがオンの状態でも更新されません。

3.5.5 リソースコンフィギュレータの定義情報

リソースコンフィギュレータのツリーペインの DevLabelVar をクリックすると，FCN/FCJ で動作中のプロジェクトに含まれるデバイスラベル変数の一覧が表示されます。

リソースコンフィギュレータから定義情報をダウンロードする際，DevLabelVar の情報とリソースコンフィギュレータの定義情報をチェックし，異なっている場合，メッセージエリアに WARNING として表示されます。

・デバイスラベル変数が定義されていない

FCN/FCJ で動作中のプロジェクトに含まれるデバイスラベル変数が，リソースコンフィギュレータ上で定義されていない場合，次の WARNING が表示されます。

例) WARNING=0026:IEC 制御アプリケーションで使用しているデバイスラベル変数（AI001）が I/O モジュールにアサインされていません。

反対に，リソースコンフィギュレータ上で定義されているデバイスラベル変数が，FCN/FCJ で動作中のプロジェクトに含まれていなくても，WARNING は表示されません。

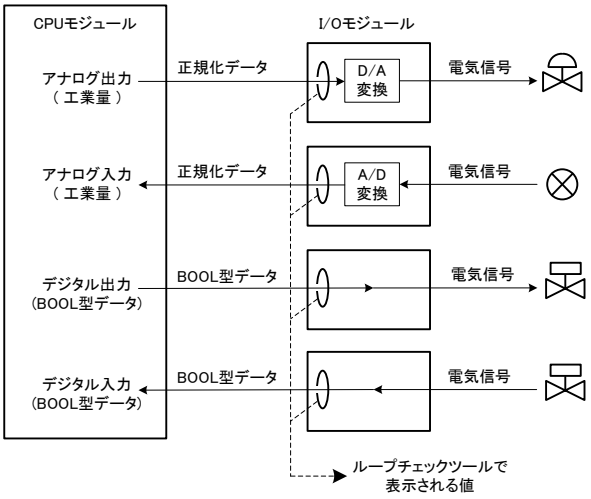
・デバイスラベルの信号種別が一致しない

FCN/FCJ で動作中のプロジェクトのデバイスラベル変数の信号種別と，リソースコンフィギュレータ上の信号種別が異なる場合，次の WARNING が表示されます。

例) WARNING=0027:IEC 制御アプリケーションで使用しているデバイスラベル変数（DI001）のデータ型（I_Sts）が I/O モジュールの信号種別と合いません。IOM=1（AI/AO），信号種別=AI

3.5.6　ループチェックツールの情報の解析

ループチェックツールは，I/O モジュールにおける入出力の値を表示します。
表示されるのは，出力信号は I/O モジュールでデジタルからアナログに変換（D/A 変換）される前の値であり，入力信号は I/O モジュールでアナログからデジタルに変換（A/D 変換）された後の値になります。



本来，ループチェックツールは，FCN/FCJ の I/O の配線チェック（ループチェック）を支援するツールですが，I/O に関する障害が発生した場合，このツールを用いて障害の原因の一次切り分けをすることができます。

■ ループチェックツールで値が表示されない

ループチェックツールの値が灰色の「＊」となっていて表示されない場合，以下の項目を確認します。

- **24V アナログフィールド電源**

  詳細は「3.2.1 電源モジュールの LED の状態の解析」を参照してください。

- **各端子の接続状態**

  各端子のケーブルの接続が正しいか，外れていないかを確認します。

- **アナログ入力モジュールの 2 線式/4 線式の設定**

  入力モジュールの中には，伝送器との接続を 2 線式接続と 4 線式接続に切り替え可能なモジュールがあります。この設定が正しくされているかを確認します。

- **I/O 切り離しになっていないか**

  ロジックデザイナのグローバル変数『GS_NFIO_DISCONNF』を ON にして I/O を切り離していると，CPU モジュールと I/O モジュールが通信をしなくなるため，ループチェックツールはデータを表示しません。

■ 実入出力の値とデバイスラベル変数の値が異なる

実際の入出力の値とデバイスラベル変数の値が異なる場合，次の 3 通りの状態が考えられます。

- **ループチェックツールとデバイスラベル変数の値は同じだが，実入出力の値が異なる場合。**

  リソースコンフィギュレータで出力反転を行っていることが考えられます。ループチェックツールとデバイスラベル変数の値が同じことから，前ページの図の CPU モジュールと I/O モジュールの D/A 変換，または A/D 変換までは正常と考えられます。
  原因はアプリケーション側ではないので，I/O モジュールと機器間の接続，ハード的な設定を確認します。

- **ループチェックツールと実入出力の値は同じだが，デバイスラベル変数の値が異なる場合。**

  ループチェックツールと実入力の値が同じことから，上の図の I/O モジュールの D/A 変換，または A/D 変換と機器までは正常と考えられます。
  原因は変換後のアプリケーションの設定にあるので，リソースコンフィギュレータの設定，ロジックデザイナのデバイスラベル変数定義など，アプリケーションを確認します。

- **ループチェックツール，デバイスラベル変数，実入出力の値が異なる場合**

  アプリケーションの定義/設定，ハード的な接続/設定の両方に原因があると考えられます。

3.6 メンテナンスページの情報の解析

3.6.1 Top Page の情報の解析

メンテナンスページの Top Page には，システムカードのメモリの容量と空き容量の情報が表示されます。
システムカードの空き容量は，システムカード上のブートプロジェクトの大きさ，ADLST のファイルサイズ，リテインデータのファイルサイズ，Java アプリケーションなどにより変化します。

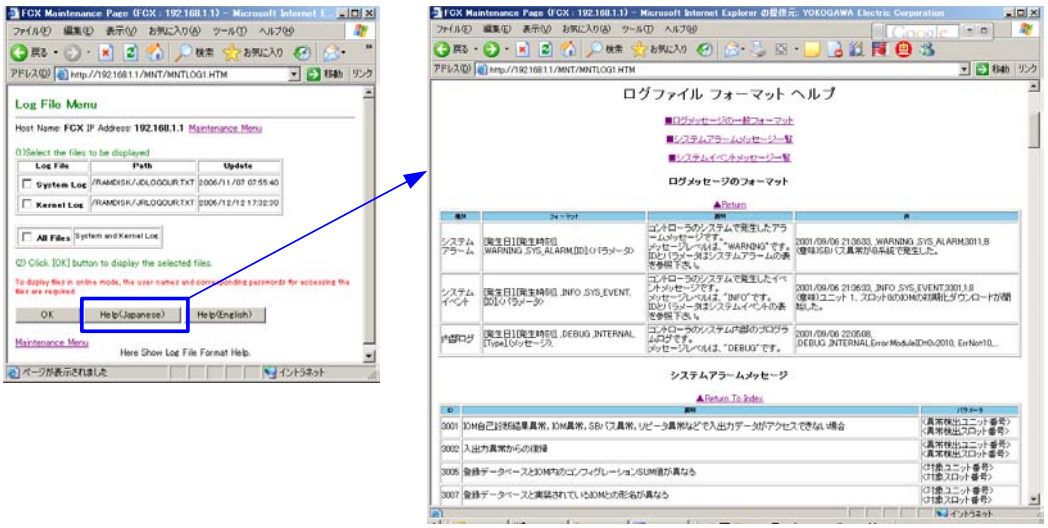
この空き容量が少ないと，システムカードのレブアップが正常に終了しません。
レブアップに失敗した場合は，空き容量を確認します。
FCN/FCJ R1～R2 では，レブアップを行うために 7MB 以上の空き容量が必要です。
FCN/FCJ R3 以降では，レブアップを行うために 10MB 以上の空き容量が必要です。

3.6.2 FCN/FCJ のログの解析

● ログの種類

メンテナンスページで表示されるログには，下記のものがあります。

- System Log
  Java アプリケーションが出力したログです。
  Java アプリケーションが動作していなければ，出力されません。
  （Java アプリケーションとは，Java を使用している APPF とユーザが作成した Java アプリケーションのことです。）
- Kernel Log
  ドライバ，実行エンジン，各種 APPF が送出したログで，情報が内部コードで表示されます。
  内部コードの詳細は，ヘルプファイルから確認できます。



● System Log の解析

System Log には，Java や APPF で発生したメッセージが，日付と時刻情報，ログの種別，内部コードと共に，記録されます。
これらの情報は 1 項目で 1 行のログになっており，発生順に記録されます。
Java で作成したアプリケーションメッセージ以外は，内部情報ですが，ある程度の想定が付く場合があります。

● Kernel Log の解析

Kernel Log には日付と時刻情報，ログの種別，内部コードが記録されます。
また，内部コードの代わりに，発生したエラーや警告の情報がテキスト情報として記録される場合もあります。
これらの情報は 1 項目で 1 行のログになっており，発生順に記録されますので，FCN/FCJ 内部で何が発生したかを時系列に沿って解析します。

● 内部コードの解析

内部コードの解析を以下の例で説明します。
以下の例において，各行の右または下の「＊」以下に，オンラインヘルプにある内部コードの情報を記述します。

例 1： 08:23:29,*,INFO ,STARDOM,JEROS(JRS:R2.04.12) starting in Online state

```
08:23:29,*,INFO ,STARDOM,JEROS(JRS:R2.04.12) starting in Online state
                                        ＊FCN/FCJ の電源が ON となると，このログが記録されます。
08:23:34, ,INFO    ,SYS_EVENT,3408,AUTO,
                                        ＊冗長化システムで制御･待機間のメモリ等値化処理を開始した。
08:23:34, ,INFO    ,SYS_EVENT,3406,     ＊制御･待機間のファイルシステムの等値化処理を開始した。"
08:24:20, ,INFO    ,SYS_EVENT,3407,     ＊ファイルシステム等値化処理が完了。
08:24:21, ,INFO    ,SYS_EVENT,3409,     ＊メモリ等値化処理が完了。
08:24:21, ,INFO    ,SYS_EVENT,3405,RIGHT,      ＊APC 終了後，待機側（右側）CPU の動作開始時。
08:24:26, ,INFO    ,SYS_EVENT,3402,LEFT,       ＊制御側 CPU（左側）の動作開始。
08:24:26, ,INFO    ,SYS_EVENT,3509,     ＊ワインドアップ開始
08:24:36, ,INFO    ,SYS_EVENT,3510,     ＊ワインドアップ終了
08:24:36, ,WARNING ,SYS_ALARM,3218,     ＊制御機能起動（Warm Start）
```

解説：08:23:29 に二重化 FCN の電源が ON になり，APC を開始し，08:24:36 左側 FCN が制御側として制御を開始しています。この時，ウォームスタートが行われていますが，これはブートプロジェクトがシステムカードからメモリ上にロードされ，ウォームスタートされた記録です。
これらのことから，電源が ON による立ち上げ処理が正常に終了したと判断できます。

例 2： 09:58:29, ,WARNING ,SYS_ALARM,3219, ＊制御機能停止

```
09:58:35, ,WARNING ,SYS_ALARM,3001,1,7,     ＊IOM 自己診断異常（UNIT1 ,SLOT7）
09:58:36, ,WARNING ,SYS_ALARM,3001,1,6,     ＊IOM 自己診断異常（UNIT1, SLOT6）
09:58:40, ,INFO    ,SYS_EVENT,3501,         ＊制御アプリケーションロード
09:58:44, ,INFO    ,SYS_EVENT,3503,         ＊ブートプロジェクトダウンロード
09:59:08, ,INFO    ,SYS_EVENT,3509,         ＊ワインドアップ開始
09:59:10, ,WARNING ,SYS_ALARM,3002,1,6,     ＊入出力復帰（UNIT1, SLOT6）
09:59:10, ,WARNING ,SYS_ALARM,3002,1,7,     ＊入出力復帰（UNIT1, SLOT7）
09:59:18, ,INFO    ,SYS_EVENT,3510,         ＊ワインドアップ終了
09:59:18, ,WARNING ,SYS_ALARM,3216,         ＊制御機能起動（Cold Start）
```

解説：09:58:29 に制御機能が停止し，それにより，FCN と I/O モジュールの通信が停止し，I/O モジュールの異常が発生しています。
ただし，その直後に制御アプリケーションとブートプロジェクトがダウンロードされていることから，この制御機能停止はダウンロードによるものであり，問題ないと判断できます。
また，ダウンロード後，FCN/FCJ はコールドスタートされていますので，リテインデータはすべて初期化されています。
この操作以後，『設定したリテインデータが消失（初期化）してしまった』という問題が発生した場合，この時のコールドスタートが原因と考えられます。

● Kernel Log のテキスト情報の解析

Kernel Log のテキスト情報は，日付情報，ログの種別と発生個所，メッセージから構成されています。

日付情報：	ログが記録された FCN/FCJ の日付と時刻が記録されます。
ログの種別：	ERROR，INFO などのログの種別が記録されます。
ログの発生個所：	FCN/FCJ のどこでログの事象が発生したかを記録します。
メッセージ：	ログの内容をテキストで表示します。

例) 20**/**/** **:**:**, ,ERROR ,ERROR,MID=0x201a,ErrNo=0x7,File=
SdcsPcException.c,Line=308,Msg=Exception(Watchdog) in Task0, Ignore

このログにおいて，それぞれ以下の様に分けられます。

日付情報：	20**/**/** **:**:**
ログの種別：	ERROR
ログの発生個所：	ERROR,MID=0x201a,ErrNo=0x7,File=SdcsPcException.c,Line=308
メッセージ：	Msg=Exception(Watchdog) in Task0, Ignore

メッセージには，その事象の内容のほかに，事象が発生したタスクの名と，事象発生時の該当タスクの動作が記録されます。
上の例では，Task0 で WDT エラーの発生を記録しています。
また，Ignore が記録されていますので，WDT エラー発生でも Task0 を停止せず，制御を継続していたことがわかります。
反対に，ログの事象発生により該当タスクを停止した場合は，Ignore ではなく，Terminate Task が記録されます。

テキスト情報のログを解析する場合，メッセージの内容だけでなく，そのログの前後に記録されているのログの内容も確認します。

● 代表的なテキスト情報のログ

代表的なテキスト情報のメッセージの内容と，その詳細を記述します。

表　代表的なテキスト情報のログ

メッセージの内容	詳　細	備考
STARDOM,JEROS(JRS:R2.04.12)starting in Online state(Option:None)	FCN/FCJ の電源が ON になると，このログが記録されます。	
Msg=Exception(Watchdog) in Task0, Ignore Msg=Exception(Watchdog) in Task0, Terminate Task	Task0 で WDT エラー発生。	3.3.3 項参照
Msg=Exception(CPU Overload) in Task0, Ignore	Task0 で CPU オーバーロード発生。	3.3.4 項参照
Msg=Exception(DIV0) in Task0, Terminate Task	Task0 の演算処理の中で 0 割発生。	
Msg=Exception(String Error) in Task0, Ignore	Task0 で文字列演算エラー発生。	
Msg=Heart beat timeout has occurred (chan=1)	FCN/FCJ 間通信でエラーが発生している。	3.8.4 項参照
Msg=MISCMP Task0:200c0200:0/80 100 1060101 0 1 0	二重化 FCN において制御側と待機側のアナログ・デジタルの出力結果が異なる。	3.8.5 項参照

3.6.3 FCN/FCJ の日付と時刻情報の解析

● FCN/FCJ の内部時計の動作と誤差

FCN/FCJ の内部の時計は CPU モジュールのバッテリーにより動作していますので，FCN/FCJ の電源が OFF の状態でも動作を継続します。
また，この時計の誤差は-17.5～+12.0 秒/日ですので，正確な時刻の情報が必要な場合は，時刻同期を行ってください。

● FCN/FCJ の時計とタイムゾーンの設定の関係

FCN/FCJ の内部時計の時刻は，世界標準時（GMT）として扱われます。
FCN/FCJ のタイムゾーンを日本標準時（JST）とすると，時計の時刻自体を+9 時間ずらすのではなく，時刻はそのままで，時刻に 9 時間足した計算値を FCN/FCJ の時刻として表示します。
例えば，FCN/FCJ のタイムゾーンが JST で，表示時刻が 09:00:00 であった場合，FCN/FCJ の内部の時計の時刻は GMT に相当する 00:00:00 となっています。
また，JST の設定での表示時刻が 09:00:00 の状態で，タイムゾーンを CTT（+8 時間）に変更すると，内部時計は 00:00:00 のままですので，表示時刻は 08:00:00 となります。

● タイムゾーンを GMT から変更せず，時刻を合わせた場合の問題

FCN/FCJ の時刻を各地域の標準時に合わせる場合，デフォルトの GMT から使用するタイムゾーンに変更し，その後，各地域の標準時で時刻を設定します。
タイムゾーンの変更をせずに時刻を合わせた場合，システムカードの交換により，表示時刻に問題が生じる場合があります。

例) タイムゾーンが GMT で，時刻を JST に合わせた場合

タイムゾーンを JST に変更せず，GMT のままで JST の時刻を設定すると，内部時計は GMT に+9 時間した時刻を，GMT 時刻として持つことになります。
この状態の FCN/FCJ でも，FCN/FCJ のタイムゾーンの設定が『GMT』であれば，タイムゾーンに合わせるための時刻の補正は行われませんので，問題はありません。
この FCN/FCJ のシステムカードを交換し，システムカードに保存されたタイムゾーンの設定が JST であった場合，FCN/FCJ の内部時計の GMT 時刻に+9 時間された時刻が表示されます。
しかし，交換前にすでに内部時計が GMT に+9 時間されていますので，交換後には GMT に+18 時間された時刻が表示され，時刻が進んでしまいます。

3.7 リモートシェルツール（FcxRsh）を用いた解析

3.7.1 リモートシェルツール（FcxRsh）の説明

リモートシェルツール（FcxRsh）は PC のコマンドプロンプトより実行するツールです。
FCN/FCJ でのコマンド操作には，telnet で FCN/FCJ へのログインが必要ですが，このツールを用いることで，FCN/FCJ でのコマンド操作を PC から行うことができます。
リモートシェルツールは FCN/FCJ のレビジョンと CPU のスタイルに関係なく，すべての FCN/FCJ で使用することができます。
リモートシェルツールは FCN/FCJ のメンテナンスホームページ，または R1.40 以降のレビジョンの CD-ROM から取得してください。

3.7.2 リモートシェルツール（FcxRsh）の取得

● FCX メンテナンスホームページから取得する手順

1. FCX メンテナンスホームページを開きます。
   http://<IP アドレス>/mnt
   ユーザ名：stardom　パスワード：YOKOGAWA
2. Maintenance Menu をクリックします。
3. Download the Backup Tools を選択し，ファイルをダウンロードします。
4. ダウンロードした FCXTOOL.EXE を解凍します。
   解凍したファイルの中に FcxRsh.exe が含まれていることを確認します。

● FCN/FCJ R1.40 以降のレビジョンの CD-ROM から取得する手順

1. R1.40 以降のレビジョンの CD-ROM を PC に挿入します。画面が自動で再生される場合は終了します。
2. エクスプローラーから CD-ROM を選び，右クリック「開く」を選択します。
3. 表示されたフォルダから「Pkg_FCX」下にある Pkg_FCX.exe を実行すると，指定したインストール先に FCX というフォルダが作られます。
   このフォルダに FCXTOOL.EXE が格納されています。

   格納場所：`FCX¥Install¥jeros¥SCRIPT`

4. FCXTOOL.EXE を解凍します。解凍したファイルの中に FcxRsh.exe が含まれていることを確認します。

重　要

FCXTOOL.EXE を解凍する際，解凍先フォルダのパスの中にスペースを含まないようにしてください。
Documents and Settings ------> NG

3.7.3 リモートシェルツール（FcxRsh）の実行

コマンド実行手順を，以下の設定を例として記述します。

コマンド格納フォルダ：	C:¥tmp¥FCXTOOL
FCN/FCJ の IP アドレス：	192.168.0.1
作成する各 BAT ファイルのファイル名：	TEST.BAT

1. コマンドプロンプトを開き，コマンドを格納フォルダに移動します。
   ```
   C:¥>cd tmp¥FCXTOOL
   ```
2. コマンドを実行します。
   リモートシェルツールの後にコマンド操作を行う FCN/FCJ の IP アドレスを指定し，続けて実行するコマンドを指定します。
   ```
   C:¥tmp¥FCXTOOL> FcxRsh 192.168.0.1  <実行コマンド>
   ```
3. 指定した FCN/FCJ で実行したコマンドの結果が表示されます。

重 要

本コマンドは，CPU の動作に影響を与えますので，運転中の使用は推奨できませんが，どうしても情報収集の必要がある場合は，代替機などでテストした後に，使用してください。Windows のコマンドプロンプトなどより FcxRsh を直接使うと，入力ミスなどでトラブルを起こす危険があるので，必ず BAT ファイルを作成し，Debug 機などで事前にテストした後に，実行してください。また，コマンド実行中に「コントロール C」などの操作を行うと，FCN/FCJ で一時的に高負荷になりますので，本操作は行わないでください。

以下に具体的なリモートシェルツールの実行例を紹介します。

● FCN/FCJ からほかの FCN/FCJ に ping を実行する

リモートシェルツールを用いて，FCN No.1（IP：192.168.0.1）から FCN No.2（IP：192.168.0.2）へ 3 回 ping を実行します。（必ず回数を指定してください。）

1. メモ帳などで，以下のコマンドを記述します。
   ```
   FcxRsh 192.168.0.1 ping ¥"192.168.0.130¥",3
   ```
   リモートシェルツールの後にコマンド操作を行う FCN/FCJ の IP アドレスを指定し，続けて実行するコマンドを指定します。
   この例では ping を 3 回実行します。
   そして，FcxRsh のコマンドが入っている下記のフォルダに拡張子を“bat”としてリモートシェルツールを格納したフォルダに TEST.BAT として保存します。
   ```
   C:¥tmp¥FCXTOOL
   ```
2. 作成したコマンドを実行します。
   ```
   C:¥tmp¥FCXTOOL>TEST.BAT
   ```

3. 指定した FCN/FCJ で実行したコマンドの結果が表示されます。

```
TEST.BAT
ping "192.168.1.2",3
PING 192.168.1.2: 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0. time=0. ms
64 bytes from 192.168.1.2: icmp_seq=1. time=0. ms
64 bytes from 192.168.1.2: icmp_seq=2. time=0. ms
----192.168.1.2 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
value = 0 = 0x0
logout
```

● Ethernet 通信をトレースする

FCN/FCJ が行っている Ethernet 通信の中で，FCN/FCJ 間通信を除く各種通信の内容を，リモートシェルツールを用いてトレースします。

重 要

通信トレースコマンドを実行すると，FCN/FCJ の CPU 負荷率が上昇しますので，コマンドを実行する場合は十分注意してください。

- トレースコマンド

通信のトレースには，SdcsPlcComFbTrace コマンドを使用します。
このコマンドを，リモートシェルツールを用いて FCN/FCJ 上で実行させ，トレース結果を PC 上に表示します。
その際，モード，通信チャンネル，トレース回数を引数として渡します。

・ 引数 1：モード

1: Text Trace ON	→ FCN/FCJ がクライアントの場合
2: Bin Trace ON	→ FCN/FCJ がクライアントの場合
3: Text&Bin Trace ON	→ FCN/FCJ がクライアントの場合
11: ModTcpSvrTrace ON	→ FCN/FCJ がサーバの場合

・ 引数 2：通信チャンネル
FCN/FCJ がクライアントの場合：通信の OPEN POU の CH 端子番号を指定します。
FCN/FCJ がサーバの場合：通信の SENDRECV POU の実行順を通信チャンネルの番号として指定します。

・ 引数 3：トレース回数
通信をトレースする回数を指定します。

・ **トレースコマンドの実行**

1. トレースコマンドを記述したテキストファイルを作成します。
   例) バイナリモード，通信チャンネル 1，トレース回数 30 回の場合

```
SdcsPlcComFbTrace 2,1,30
#
taskDelay 500
#
```

   テキストファイル名は任意の名前でかまいませんが，リモートシェルツールと同じフォルダに保存します。例として Trace_Ether.txt として説明します。

2. BAT ファイルを作成してリモートシェルツールを実行します。
   リモートシェルツール（FcxRsh）の後に–f オプションでファイル名（Trace_Ether.txt）と，その後にトレースする FCN/FCJ に IP アドレスを指定します。
   例) ツールの解凍フォルダを C:¥temp¥FCXTOOL，FCN/FCJ の IP アドレスが 192.168.0.2 の場合の BAT ファイルの中身

```
FcxRsh -f Trace_Ether.txt 192.168.0.2
```

3. BAT ファイルを実行した時のトレース結果が表示されます。

```
TEST.BAT
SdcsPlcComFbTrace 2,1,30
value = 0 = 0x0
taskDelay 500
CH= 1 S=12   OK : 00000000 00060101 00000014
CH= 1 R=12   OK : 00000000 00060101 03000000
CH= 1 S=12   OK : 00000000 00060102 00000014
                :
                :
CH= 1 S=12   OK : 00000000 00060101 00000014
CH= 1 R=12   OK : 00000000 00060101 03000000
value = 0 = 0x0
logout
```

● シリアル通信をトレースする

FCN/FCJ が行っている各種シリアル通信の内容を，リモートシェルツールを用いてトレースします。

重 要

通信トレースコマンドを実行すると，FCN/FCJ の CPU 負荷率が上昇しますので，コマンドを実行する場合は十分注意してください。

・トレースコマンド

通信のトレースには，SdcsRsmFbTrace コマンドを使用します。
このコマンドを，リモートシェルツールを用いて FCN/FCJ 上で実行させ，トレース結果を PC 上に表示します。
その際，モード，通信ポート名を引数として渡します。

・ 引数 1：モード
1: Text　Trace ON
2: Bin　Trace ON
3: Text&Bin Trace ON

・ 引数 2：通信ポート名
COM1：FCN のシリアル通信ポート，または FCJ のシリアル通信ポート 1 の通信をトレースする場合。
COM2：FCJ のシリアル通信ポート 2 の通信をトレースする場合。
その他：RS 通信モジュールを使用する場合，リテインデータで設定したポート名を設定します。

・トレースコマンドの実行

1. トレースコマンドを記述したトレーススタート用のテキストファイルを作成します。
例）バイナリモード，FCN のシリアル通信ポートの場合

```
SdcsRsmFbTrace 2,"COM1",30
#
taskDelay 500
#
```

テキストファイル名は任意の名前でかまいませんが，リモートシェルツールと同じフォルダに保存します。例として Trace_Serial_Start.txt として説明します。

2. 起動用の TEST.BAT ファイルを作成します。その中身は下記のとおりです。

```
FcxRsh -f Trace_Serial_Start.txt 192.168.0.2
```

3. コマンドを終了させる END.BAT ファイルを作成します。その中身は下記のとおりです。

```
FcxRsh SdcsRsmFbTrace
#
```

4. 起動用の TEST.BAT ファイルを実行します。
   C:¥temp¥FCXTOOL>TEST.BAT
   トレース結果が表示されます。

```
TEST.BAT
 SdcsRsmFbTrace 2,"COM1",30
 value = 0 = 0x0
 -> COM1     S=8    OK : 01010000 00143c05
COM1     R=8    OK : 01010300 00003c4e
COM1     S=8    OK : 01020000 001ef802
COM1     R=9    OK : 01020400 000000fb e2
COM1     S=8    OK : 01030000 00044409
COM1     R=13   OK : 01030800 00000000 00000095 d7
COM1     S=8    OK : 01040000 00067008
COM1     R=17   OK : 01040c00 00000000 00000000 00000095 b7
COM1     S=8    OK : 01050064 ff00cde5
COM1     R=8    OK : 01050064 ff00cde5
COM1     S=8    OK : 01060064 04d24a88
COM1     R=8    OK : 01060064 04d24a88
    :
    :
```

5. コマンドを終了させる END.BAT ファイルを実行し，FCN/FCJ 上のトレースコマンドを停止させます。
   例) ツールの解凍フォルダを C:¥temp¥FCXTOOL，FCN/FCJ の IP アドレスが 192.168.0.2 の場合
   C:¥temp¥FCXTOOL>END.BAT

6. SdcsRsmFbTrace の Usage が表示されれば，FCN/FCJ 上のトレースコマンドは停止しています。

```
END.BAT
--------------------------------------------------------------------------------
Usage : SdcsRsmFbTrace mode, "PortName", NumberOfTrace, timeStamp, errorStop
--------------------------------------------------------------------------------
mode    :   1: Text     Trace ON
            2: Bin      Trace ON
            3: Text&Bin Trace ON
                :
                :
--------------------------------------------------------------------------------
value = 0 = 0x0
logout
```

3.8 FCN/FCJ の障害の解析事例

FCN/FCJ で発生した障害の種類によっては，複数の情報を総合して解析をする必要があります。以下にその事例を紹介します。
また，これらの事例は解析手法を紹介するための参考例であり，これらがすべてではありません。

3.8.1 リテインデータに関する障害の解析事例

● 発生した障害

・リテインデータが初期化された。
・リテインデータが古い値に戻ってしまった。

● 必要な情報

	情報	参照先
1.	CPU モジュールの基本設定	3.5.1 項
2.	リテインデータの日付情報	「システム解析ガイド（情報収集編）」（TI 34P02K03-02） 3.6.3 項
3.	FCN/FCJ のログ情報	3.6.2 項

● 解析事例

まず，CPU モジュールの基本設定より，『リテインデータのハードバックアップを有効にする』のチェックの有無を確認し，リテインデータが SRAM 上に保持されているかを確認します。

- リテインデータが SRAM 上に保持されている場合（FCN/FCJ のデフォルト設定）

FCN/FCJ の電源 OFF/ON やウォームスタートでは，SRAM 上のデータがそのまま使用されます。
このデータが初期化，または古くなっていたということは，何らかの理由でリテインデータが初期化されたか，システムカードからリストアされたかのいずれかと考えられます。

- リテインデータが SRAM 上に保持されていない場合

FCN/FCJ の電源 OFF/ON やウォームスタートで，リテインデータはシステムカードからリストアされます。
このデータが初期化，または古くなっていたということは，システムカードにリテインデータがセーブされていなかったか，セーブされたデータが古かったことが考えられます。

次に FCN/FCJ のログ情報に，下記の 5 個の内部コードが残されているかを確認し，FCN/FCJ がどのような状態になったかを，時系列的に解析します。
「3.6.2 項 FCN/FCJ のログの解析」を参照してください。

3505　リテインデータセーブ開始
3506　リテインデータセーブ終了
3507　リテインデータリストア開始
3508　リテインデータリストア終了
3216　制御機能起動（Cold Start）

その結果，3507 と 3508 のリテインデータリストアのログが記録されていれば，そのタイミングで SRAM 上のリテインデータに，システムカードのデータが上書きされています。
また，FCN/FCJ が 3216 のコールドスタートされていると，そのタイミングでリテインデータは初期化されています。

リテインデータのリストアが確認されたならば，リテインデータの日付情報から，そのデータがいつ保存されたかを確認します。

3.8.2　入出力信号に関する障害の解析事例

● 発生した障害

・ FCN/FCJ の入出力のデータと，実入出力のデータが一致しない。
・ IOP，OOP が発生している。

● 必要な情報

	情報	参照先
1.	I/O モジュールの状態表示 LED の状態	3.2.4 項
2.	電源モジュールの LED の状態	3.2.1 項
3.	システム定義のグローバル変数の情報	3.4.1 項
4.	リソースコンフィギュレータの定義情報	3.5.5 項
5.	ループチェックツールの情報	3.5.6 項

● 解析事例

まず，I/O モジュールの状態表示 LED の状態から，I/O モジュールが正常に動作しているかを確認します。（「3.2.4　I/O モジュールの状態表示 LED の解析」参照）
同様に，電源モジュールの LED の状態から，24V アナログフィールド電源を必要とする I/O モジュールに電源が供給されているか，消費電流が定格の 4A 以下であるかを確認します。（「3.2.1　電源モジュールの LED の状態」参照）
システム定義のグローバル変数『GS_RAS_IOStat』の情報を確認し，I/O モジュールのステータスとエラー状態を確認します。（「3.4.1　システム定義のグローバル変数の情報」参照。）
I/O モジュールの状態に問題がないこと，WARNING がないことを確認した後，ループチェックツールを用いて，障害の原因がハード的な接続に問題があるのか，アプリケーションの設定などの問題なのかの一次切り分けを行います。（「3.5.6 ループチェックツールの情報の解析」参照。）
一次きり分け後，問題個所を修正します。

3.8.3 時間計測に関する障害の解析事例

● 発生した障害

FCN/FCJ でタイマファンクションブロック（TP, TON, TOF）や NPAS_TM を使用していて，時間が正しく計測されない。

● 必要な情報

	情報	参照先
1.	制御タスクの実行時間	3.3.1 項
2.	WDT エラーの発生状況	3.3.3 項
3.	制御アプリケーション	―

● 解析事例

タイマファンクションブロック（以下，TM FB）や NPAS_TM は，制御タスクの制御周期と，TM FB や NPAS_TM が起動された回数から時間を計測します。

例えば，制御周期が 100msec の制御タスクで TON が起動されると，1 回につき制御周期（=100msec）が経過時間に加算されます。
プリセット時間を 1 秒に設定した場合，タイムアップとなるのは動作開始から 1 秒後ではなく，TON が 10 回起動（1 秒=10 回＊100msec）されるとプリセット時間が経過したと判断し，タイムアップとなります。

1. 制御タスクの実行時間の解析

経過時間の計測が正しく行われない場合，まずは制御タスクの実行時間を確認します。（「3.3.1　制御タスクの実行時間」参照）
その結果，実行時間＞制御周期となっていると，時間計測は正しく行われません。
上の例において，制御タスクは 1 秒間に 10 回が起動されます。この中の 1 回の実行時間が 110 msec（実行時間＞制御周期）となると，1 秒間に 9 回しか起動されず，経過時間は 900 msec と判断されます。
タスクが 10 回起動されるには 1.1 秒必要になり，プリセット時間以上の時間がかかってしまいます。（「3.3.3　ウォッチドックタイムエラー」参照）
また，WDT=制御周期の設定であれば，実行時間＞制御周期となるとログファイルに記録が残されますので，解析の情報となります。

2. アプリケーションの解析

制御タスクの実行時間に問題がない場合，制御アプリケーションの確認を行います。
制御アプリケーションにおいて，IF 文や CASE 文での条件判断，JUMP 処理，RETURN 処理などにより，TM FB や NPAS_TM の起動がスキップされることがあります。この場合，TM FB や NPAS_TM に起動がかからないため，時間計測が正しく行われません。
また，FOR NEXT 文により，1 回の制御周期において TM FB や NPAS_TM が複数回起動されることもあります。この場合，プリセット時間よりも早くタイムアップします。

3.8.4 Heart beat timeout エラー

● 発生した障害

FCN/FCJ のログファイルを見たところ，FCN/FCJ 間通信のエラーを示す Heart beat timeout エラーが記録されていた。

● エラーの詳細

FCN/FCJ 間通信でエラーが発生すると，次の様な Heart beat timeout エラーがログファイルに記録されます。

例) 200*/**/** **:**:** ,ERROR,ERROR,MID=0x2025,ErrNo=0,File=Sdcs_commfb.c, Line=1681,
Msg=Heart beat timeout has occurred (chan=1).(sent=2674433, now=2674753)
200*/**/** **:**:** ,ERROR,ERROR,MID=0x2025,ErrNo=0,File=Sdcs_commfb.c, Line=4411,
Msg=doHBTout()/resetC() called. (chan=1) P=/inet/ -ip 192.168.1.1 -p 20001 -srv

FCN/FCJ 間通信を行っている FCN/FCJ は，通信相手からのデータ到着を待っています。通信相手からのデータが 1.5 秒間来なかった場合，FCN/FCJ は通信相手の状態の確認のため Heart beat 通信を送ります。
通常は HeartBeat 通信の応答はすぐに送信元の FCN/FCJ に戻ってきますが，何らかの原因で応答が 2 秒以上ないと Heart beat timeout エラーが発生します。
ログの内容から，エラーの発生した通信相手の FCN/FCJ の IP アドレスが ‘192.168.1.1’であることがわかります。
また，一度 Heart beat timeout エラーが発生すると，FCN/FCJ 間通信が回復して再度通信エラーとならない限り，次の Heart beat timeout エラーは発生しません。
例えば，FCN/FCJ 間をつなぐ制御ネットワークが断線すると Heart beat timeout エラーは出力されますが，断線の状態でエラーが出力され続けることはありません。
次に出力されるのは，制御ネットワークが復旧し，再度断線したタイミングです。

● 必要な情報

	情報	参照先
1.	FCN/FCJ のログ	3.6.2 項
2.	CPU 負荷率	3.3.2 項

● 解析事例

Heart beat timeout エラーの要因として，次の項目が考えられます。

- 通信経路の経路の異常

- LAN ケーブルの断線，HUB の故障など，物理的な異常による通信エラー。
- ノイズなどによる通信エラー。
- ネットワーク負荷上昇による通信エラー。

- 相手 FCN/FCJ の異常

- 通信相手の FCN/FCJ がダウンしたことによる通信エラー。

- 相手 FCN/FCJ の CPU の高負荷

・ 制御アプリケーションの処理は FCN/FCJ 間通信より優先順位が高いため，FCN/FCJ の CPU 負荷率が高くなると FCN/FCJ 間通信の処理ができなくなり，通信エラーとなります。

次に FCN/FCJ のログファイルから，Heart beat timeout エラーの発生が頻発しているか，単発なのかを確認します。
エラーが単発であれば，一過性のネットワークや CPU の負荷上昇による通信エラーの可能性が考えられます。
Heart beat timeout エラーが頻発していれば，下記の様に FCN/FCJ 間通信の発生と復帰が繰り返されています。
一度 Heart beat timeout エラーが発生すると，FCN/FCJ 間通信が回復して再度通信エラーとならない限り，次の Heart bear timeout エラーが発生しません。
例えば，FCN/FCJ 間をつなぐ制御ネットワークが断線すると Heart beat timeout エラーは出力されますが，制御ネットワークが断線の状態で Heart beat timeout エラーが出力され続けることはありません。次に出力されるのは，制御ネットワークが復旧し，再度断線したタイミングです。
このことから，Heart beat timeout エラーが頻発している場合，恒常的なネットワークや CPU の負荷上昇，ノイズの発生，LAN ケーブルや HUB の不調の可能性が考えられます。

3.8.5 MISCMP エラー

● 発生した障害

CPU を二重化している FCN のログファイルを見たところ，制御側と待機側の出力が異なっていることを示す MISCMP エラーが記録されていた。

● エラーの詳細

二重化 FCN において，制御側と待機側のデバイスラベルの出力の値を比較しています。
通常では制御側と待機側の出力は同じになりますが，何らかの原因で異なる値が出力されると，MISCMP エラーがログファイルに記録されます。

例) 200*/**/** **:**:** , ,ERROR,ERROR,MID=0x2007,ErrNo=0x20080013,File=SdcsRedcyTskSyncQ.c, Line=1178,Msg=MISCMP Task0:200c0200:2c/80 100 1060101 0 1 0

出力が異なる場合，制御側の値が実際に出力され制御は継続します。
しかし，この状態で制御側 CPU がフェイルし，制御権が待機側に切り替わると，出力が急変します。

例えば，両側 CPU が正常な状態で，ある DO の出力が制御側は ON，待機側は OFF となっていると，FCN からは制御側 CPU の値の ON が出力されます。
この状態で制御側 CPU がフェイルし，制御権が待機側 CPU に切り替わると，DO の出力は OFF となります。
また，旧待機側であった CPU が制御側になるため，内部の情報が更新されて DO の出力がすぐ ON に戻ることも考えられます。
この様に，MISCOMP エラーが発生している状態で CPU の制御権が切り替わると，出力値が急変したり，DO が一瞬だけ反転する場合が考えられ，制御に影響を及ぼす可能性があります。

● 必要な情報

	情報	参照先
1.	FCN のログ	3.6.2 項
2.	ロジックデザイナのプロジェクト	―

● 解析事例

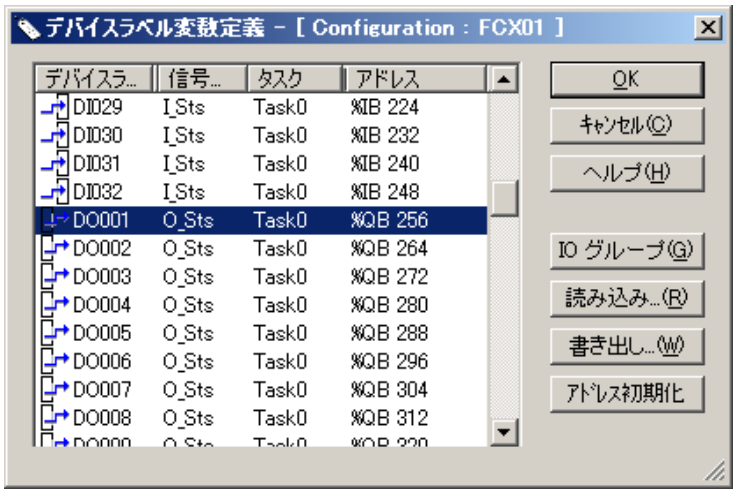
FCN のログに記録されている情報から，MISCMP エラーが発生しているデバイスラベルを特定し，必要に応じてアプリケーションを修正します。
上記の MISCMP エラーの例において，『Msg=MISCMP Task0:200c0200:0/80 100 1060101 0 1 0』の情報があり，『100 1060101 0 1 0』の部分から，MISCMP エラーの発生個所を特定します。

『100 1060101 0 1 0』は次の 5 個の情報に分けることができます。

100	1060101	0	1 0
第 1 情報	第 2 情報	第 3 情報	第 4，5 情報

- **第 1 情報**

該当デバイスラベルのアドレスを 16 進数で表示します。
上記の第 1 情報の 100（16 進数）=256（10 進数）から，デバイスラベルのアドレスが 256 であることがわかります。
デバイスラベルのアドレスはロジックデザイナのデバイスラベル定義に表示されています。このアドレスはデバイスラベルの種類に関係なく，先頭から連番で振られますので，アドレスが 256 であるデバイスラベルを探します。
下の図の例では 256 のアドレスは DO001 であり，このデバイスラベルで MISCMP エラーが発生しています。



- **第 2 情報**

該当デバイスラベルのハードの実装情報を表示します。
MISCMP エラーが発生しているデバイスラベルの情報がユニット，スロット，ポート，チャンネルの順に記録されています。

ユニット：	1：制御ユニット 2：増設ユニット 1 3：増設ユニット 2
スロット：	I/O モジュールが実装されているスロット番号
ポート：	1 で固定
チャンネル：	デバイスラベルの実装チャンネルを 16 進数で表示

上記の第 2 情報『1060101』では，次の様になります。

1	06	01	01
ユニット	スロット	ポート	チャンネル

これにより，制御ユニットの 6 スロットの I/O モジュールのチャンネル 1 で MISCMP が発生していることがわかります。

- **第 3 情報**

内部情報ですので解析には使用しません。

- **第 4，5 情報**

自 CPU 側の出力値と相手側 CPU の出力値を，自 CPU，相手側 CPU の順で表示します。自 CPU とは，ログ情報を表示している CPU のことで，通常は制御側 CPU になります。
上記の第 4，5 情報『1 0』では，次の様になります。

1	0
自 CPU の出力	相手側 CPU の出力

これにより，自 CPU（制御側 CPU）の出力は 1（=ON），相手側（待機側 CPU）の出力が 0（=OFF）であることがわかります。

- **結果**

以上の解析より，上記の例の『100 1060101 0 1 0』は，

デバイスラベルのアドレス：	%QB256 の DO001
実装位置：	制御ユニット，6 スロット，チャンネル 1

のデバイスラベルにおいて，

制御側 CPU 出力：	ON
待機側 CPU 出力：	OFF

になっているので MISCMP エラーが発生していることがわかります。

3.8.6　FCN/FCJ と PC 間の通信ができなくなった。

● 発生した障害

FCN/FCJ の接続された PC において，ASTMAC/VDS とのデータ送受信，ロジックデザイナのデバッグモードやプロジェクトコントロールダイアログ，リソースコンフィギュレータでのアップロード，Internet Explorer でのメンテナンスページの表示，などの各通信ができなくなった。

事例 1

FCN/FCJ と PC 間の各種通信が行われないうえ，制御も行われていない。
ただし，FCN/FCJ と PC 間の ping コマンドは通る。

事例 2

FCN/FCJ の IP アドレスを設定（変更）したところ，PC 間の各種通信が行われなくなった。制御は行われているようである。
ただし，FCN/FCJ と PC 間の ping コマンドは通らない。

● 必要な情報

	情報	参照先
1.	電源モジュールの LED の状態	3.2.1 項

注：　FCN/FCJ と PC 間の通信ができないため，そのほかの情報収集はできない。

● 事例 1 の解析

結論として，この現象は 3.3.4 項で記述されている CPU オーバーロードが発生しているか，CPU の負荷がそれに近い状態になると発生します。
FCN/FCJ の LED の状態，FCN/FCJ と PC 間の ping コマンドが通ることから，
FCN/FCJ 自体は動作しています。
動作状態が正常にもかかわらず各通信ができなくなっている原因として，
FCN/FCJ の制御アプリケーションの負荷が高くなり，CPU の空き時間がなくなっていることが考えられます。
上述の各通信は制御アプリケーションよりプライオリティが低いため，CPU の空き時間で行われます。制御アプリケーションの負荷が高く，CPU 空き時間がなくなると通信ができなくなり，タイムアウトなどを起こします。
また，PC からの ping コマンドに対する対応は，制御アプリケーションよりもプライオリティが高く設定されています。そのため，CPU の空き時間がなくても，ping コマンドは通ります。

CPU の空き時間がなくなる原因として，次の要因が考えられます。

- 制御タスクより優先順位の高いタスク処理が継続して行われた。
- 制御タスクの周期が短い。
- 制御アプリケーションの繰り返し処理が終わらなくなった。
  （詳細は「3.3.4　CPU オーバーロード」を参照。）

- **制御タスクより優先順位の高いタスク処理が継続して行われた。**

  制御タスクより優先順位の高いタスク処理が継続する代表的な原因として，Ethernet 上のブロードキャスト通信があげられます。（「3.4.3　ロジックデザイナのログ情報の解析」項参照）
  この要因を確認するため，該当 FCN/FCJ 以外の機器を制御ネットワークから切り離し，ネットワーク上に該当 FCN/FCJ と PC だけが存在する状態にします。もしくは，PC とクロスケーブルで直結します。
  また，該当 FCN/FCJ が社内情報 LAN に接続されている場合は，その LAN からも切り離します。
  この状態で，FCN/FCJ と PC 間の通信を確認します。
  通信が復旧したら FCN/FCJ のログ情報を確認し，CPU オーバーロードエラーや WDT エラーが記録されているかを確認します。

- **制御タスクの周期が短い。**
  制御アプリケーションの繰り返し処理が終わらなくなった。

  これらは，制御タスクの設定，または制御アプリケーションに起因します。
  そこで，該当 FCN/FCJ を制御アプリケーションが動作していない状態にし，通信ができるかを確認します。
  制御アプリケーションが動作していない状態にするには，ブートプロジェクト削除ツール（FcxDeleteProject）を用いてシステムカード内のブートプロジェクトを削除し，FCN/FCJ をリブートします。これにより，FCN/FCJ は制御アプリケーションが動作しない状態で立ち上がります。
  通信ができない原因が制御タスクの設定，または制御アプリケーションにある場合は，制御アプリケーションが動作していない状態にすることにより，通信ができるようになります。
  通信が復旧したら FCN/FCJ のログ情報を確認し，CPU オーバーロードエラーが記録されているかを確認します。

  ログ情報確認後，ロジックデザイナで原因となっているアプリケーションの修正を行い，FCN/FCJ にダウンロードします。
  その際，削除してしまったブートプロジェクトにもダウンロードすることを忘れないでください。

- ブートプロジェクト削除ツール（FcxDeleteProject）の実行手順

1. ツールの取得
   3.7.1 項のリモートシェルツールと同じ手順でブートプロジェクト削除ツール（FcxDeleteProject）を取得します。
   ツールを格納するフォルダの名前にはスペースを使わないでください。

   例) Documents and Settings ------> NG

2. FCN/FCJ を一度シャットダウンの後 Reboot します。
   その後ブートモード選択状態でメンテナンスモードにて立ち上げます。

3. ブートプロジェクト削除ツールの実行
   コマンドプロンプトを開き，ブートプロジェクト削除ツールを実行します。

   ```
   c:¥<ツールの格納フォルダ>FcxDeleteProject <IP アドレス>
   ```

   ツール実行例）

   格納フォルダ：tmp¥FCXTOOL　IP アドレス：192.168.0.2 の場合
   C:¥tmp¥FCXTOOL>FcxDeleteProject 192.168.0.2
   remove "/JEROS/STARDOM/PROCONOS/BOOTFILE.PRO"
   value = 0 = 0x0
   logout

   logout が表示されてコマンド終了です。

4. FCN/FCJ の立ち上げ
   ブートプロジェクト削除ツールを実行後，リブートか電源 OFF/ON にて FCN/FCJ を立ち上げます。

● 事例 2 の解析

ping コマンドが通らないことから，FCN/FCJ の IP アドレスが書き換えられている可能性があり，その原因として，DHCP サーバによる IP アドレスの設定が考えられます。
FCN/FCJ を接続しているネットワークに DHCP サーバが存在すると，FCN/FCJ を IP アドレス設定モードに移行したタイミングで，リソースコンフィギュレータからの設定より先に，DHCP サーバから IP アドレスが設定される場合があります。
DHCP サーバからの IP アドレスは任意であるため，FCN/FCJ と PC 間の各種通信ができなくなり，ping コマンドも通らなくなります。
ただし，事例 1 と異なり，CPU オーバーロードエラーは発生していませんので，制御は行われます。

この事例の場合，FCN/FCJ を制御ネットワークから切り離し，PC と直結の状態にして IP アドレスを設定します。PC と通信ができることを確認後，制御ネットワークに接続します。

Blank Page

付録 A.　パフォーマンスデータ

付録 A.1　メモリ見積もり

● ASTMAC アプリケーションシステムの一般的なメモリ見積もり

ページ数，使用する ActiveX コントロールや市販ソフトにより変わります。メモリは多めに実装することをお勧めします。

表 A1　メモリ見積もり（単位 MB）

AP フォーム数	フルタイム	ランタイム
1～2	96～128	64～128
3～9	128～256	96～256
10～16	256～768	256～512

● トレンドコントロール（Trex2）のメモリ見積もり

- リアルタイムトレンドの場合

画面：表示領域サイズ Max:1024×768×24Bit（色）＝約 3MByte　→　①
リアルタイムトレンド使用時保持データ
ペン数（最大 8）×データ数（最大 10,800 点）×データ（8Byte）＝675Kbyte（最大）　→　②
（ペン数＋1）×表示幅ドット数（最大 1024 ドット）×4Byte＝320Kbyte（最大）　→　③
上記①＋②＋③で求められます。

- ヒストリカルトレンドの場合

画面：表示領域サイズ Max:1024×768×24Bit（色）＝約 3MByte　→　①
ヒストリカルトレンド使用時保持データ
ペン数（最大 8）×データ数（最大 655360 点）×データ（8Byte）＝約 4Mbyte（最大）　→　②
（ペン数＋1）×表示幅ドット数（最大 1024 ドット）×4Byte＝320Kbyte（最大）　→　③
上記　①＋②＋③　で求められます。

● アラームサマリのメモリ見積もり

約 500KB+最大アラーム件数×0.5 KB

● メッセージヒストリのメモリ見積もり

約 500KB+最大履歴件数×0.5 KB

付録 A.2　通信時間見積もり

● FA-M3 の通信時間見積もり

VDS/ASTMAC と FA-M3 との通信時間は，ラダーのスキャンタイム，CPU カード／上位リンクユニットカード（LE01 も含む）／表示器接続カードの有無によって影響を受け，30ms～100ms くらいかかります。
以下にデータ点数との関係を示します。なお，このデータは上位リンクアクセスとして条件の良い構成で測定したものですので注意ください。

表　VDS/ASTMAC が 1 データ収集を行う時間（FA-M3R）

1 パケットあたりのデータ点数	1 点	4 点	16 点	64 点	256 点
ビットデバイス（Ethernet）	25.5 msec	25.5 msec	25.4 msec	25.4 msec	25.0 msec
ワードデバイス（Ethernet）	25.5 msec	25.4 msec	25.2 msec	25.2 msec	-
ビットデバイス（RS）	118.0 msec	120.0 msec	126.1 msec	152.2 msec	252.3 msec
ワードデバイス（RS）	120.2 msec	126.2 msec	152.1 msec	251.2 msec	-

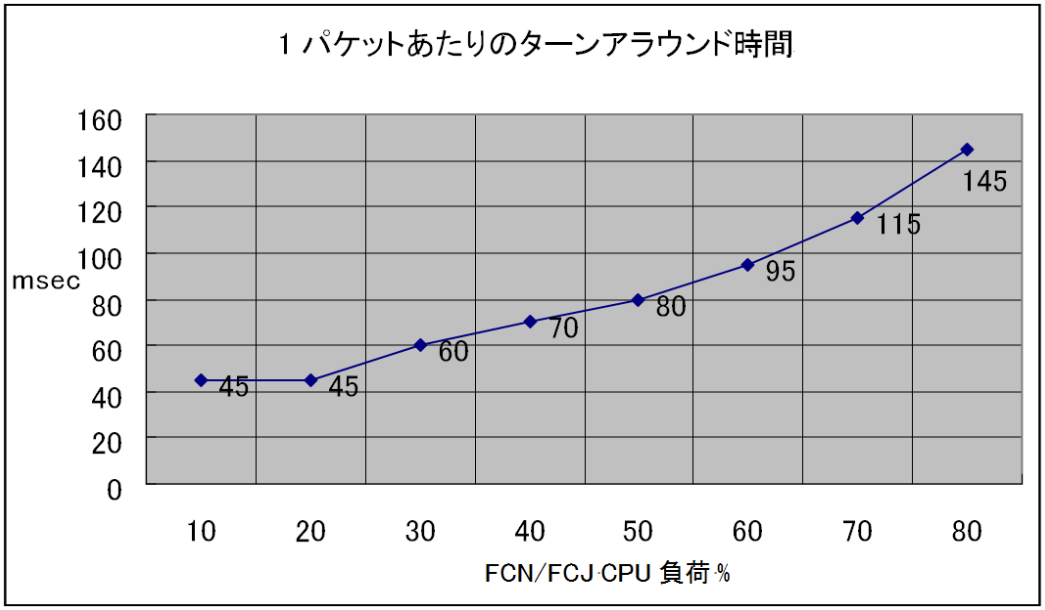
表　VDS/ASTMAC が 1 データ収集を行う時間（FA-M3R タイプ）

	498 点（ワード）	4096 点（ビット）
FA-M3R（Etehernet）	32.7 msec	30.0 msec

● FCN/FCJ の通信時間見積もり

VDS/ASTMAC と FCN/FCJ との通信時間は，FCN/FCJ の CPU 負荷に大きく依存します。1 回のアクセスに 50msec～100msec ぐらいかかると考えてください。

以下に FCN/FCJ の CPU 負荷と 1 パケットあたりのターンアラウンド時間を示します。データサーバに PAS_PID オブジェクトを 10 オブジェクト登録して実測



また，1 回の通信で 2Kbyte（2048byte）のデータアクセスが可能です。データ型によって 1 回で読み込めるデータ点数は変わります。以下に代表的な型を紹介します。

データ型	1 回に読める点数
BOOL	512 点 ※32bit 以下の型はすべて 4byte に拡張されます。
BYTE	
INT	
REAL	
CDataReal （32byte）	64 点
NPAS_PVI（124byte）(*1)	16 点
NPAS_PID（196byte）(*1)	10 点

*1： NPAS オブジェクトのバイト数は，「通常スキャン」のみの型を加算した値です。NPAS オブジェクトごとにデータ型の数は異なります。
各 NPAS オブジェクトのアクセスバイト数は，「IM 34P02D02-01 VDS エンジニアリングマニュアル」の「C5. PASPOU オブジェクト」を参照して計算してください。

1 秒あたり収集できるデータ数は以下のようになります。（参考例）

CPU 負荷別 アクセス可能データ点数/秒	20% （50msec）(*2)	40% （70msec）(*2)	60% （100msec）(*2)
PAS_PVI	320 点 (*3)	224 点	160 点
PAS_PID	200 点	140 点	100 点

*2： （）内は，1 回の通信（2Kbyte）アクセス時間
*3： 1 秒あたりのデータ数=1 回のアクセス点数×1000/1 回のアクセス時間

Blank Page

Technical Information　改訂情報

資料名称　　：STARDOM システム解析ガイド（情報解析編）
資料番号　　：TI 34P02K03-03

’00 年 4 月／初版／R2.05 以降
新規発行

’01 年 3 月／2 版／R3.01 以降
改訂

’02 年 1 月／3 版／R4.01 以降
改訂

’02 年 10 月／4 版／R4.11 以降
改訂

’06 年 01 月／5 版／R5.30，R1.70 以降
改訂
・VDS/ASTMAC システム解析ガイドを情報収集編と情報解析編に分冊

’07 年 07 月／6 版／R5.40，R1.80 以降
改訂

’09 年 03 月／7 版／R6.01，R2.01 以降
改訂
・誤記訂正

’09 年 09 月／8 版／R6.20，R2.20 以降
改訂
・誤記訂正

’11 年 03 月／9 版／R7.01，R3.10 以降
改訂
・Windows 7 対応
・誤記訂正

’12 年 06 月／10 版／R7.10，R3.20 以降
改訂
・新モジュール情報追加（NFLC121,NFLP121）

’12 年 09 月／11 版／R7.10，R3.20 以降*
改訂
・誤記訂正

*: Technical Information 記載内容と対応しているソフトウェアのリリース番号。対応する範囲は次の改訂版が発行されるまで。

■ お問い合わせについて
本書の内容に関するご質問は，下記メールアドレスにて
お願いいたします。
問い合わせメールアドレス：stardom_info@cs.jp.yokogawa.com
■ 著作者横河電機株式会社
■ 発行者 横河電機株式会社
〒180-8750 東京都武蔵野市中町 2-9-32